CAS

WANXIANG BLOCKCHAIN LABS

BITCOIN AND CRYPTOCURRENCY TECHNOLOGIES

A Comprehensive Introduction

区块链
技术驱动金融

数字货币与智能合约技术

[美] 阿尔文德·纳拉亚南 (Arvind Narayanan)　约什·贝努 (Joseph Bonneau)

爱德华·费尔顿 (Edward Felten)　安德鲁·米勒 (Andrew Miller)

史蒂文·戈德费德 (Steven Goldfeder)◎著

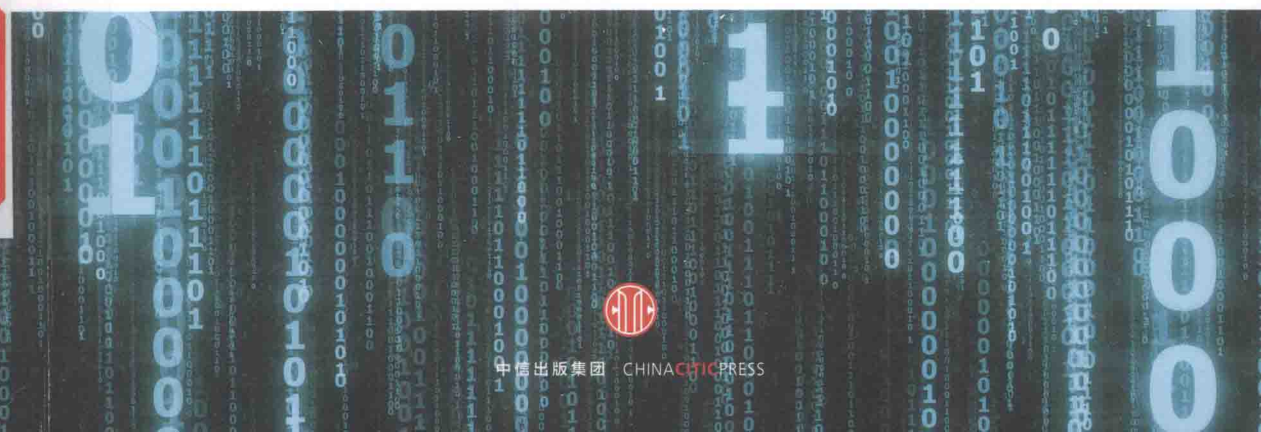
林　华　王　勇

帅　初　蔡凯龙　许余洁　李耀光　高晓婧　洪　浩◎译

解密区块链，用技术重构金融世界

谢平 中国投资公司前副总经理　**肖风** 中国万向控股有限公司副董事长　倾情作序

邢早忠 金融时报社社长　**霍学文** 北京市金融工作局局长　**刘信义** 浦发银行行长　**黄世忠** 厦门国家会计学院院长　**唐　斌** 深圳前海金融资产交易所总经理　联袂推荐



中信出版集团 CHINACITICPRESS

BITCOIN

AND

CRYPTOCURRENCY

TECHNOLOGIES

A Comprehensive Introduction

区块链

技术驱动金融

数字货币与智能合约技术

[美] 阿尔文德·纳拉亚南(Arvind Narayanan) 约什·贝努(Joseph Bonneau)
爱德华·费尔顿(Edward Felten) 安德鲁·米勒(Andrew Miller)
史蒂文·戈德费德(Steven Goldfeder)◎著
林 华 王 勇
帅 初 蔡凯龙 许余洁 李耀光 高晓婧 洪 浩◎译

中信出版集团·CHINACITICPRESS·北京

图书在版编目（CIP）数据

区块链：技术驱动金融 /（美）纳拉亚南等著；林华等译. 一 北京：中信出版社，2016.8（2016.10 重印）

书名原文：Bitcoin and Cryptocurrency Technologies: a Comprehensive Introduction

ISBN 978-7-5086-6584-9

I. ①区… II. ①纳… ②林… Ⅲ. ①电子货币－基本知识 Ⅳ. ① F830.46 ② TP3

中国版本图书馆 CIP 数据核字（2016）第 182827 号

区块链：技术驱动金融

著　　者：（美）阿尔文德 · 纳拉亚南　约什 · 贝努　爱德华 · 费尔顿　安德鲁 · 米勒　史蒂文 · 戈德费德
译　　者：林　华　王　勇　帅　初　蔡凯龙　许余洁　李耀光　高晓婧　洪　浩
策划推广：中信出版社（China CITIC Press）
出版发行：中信出版集团股份有限公司
（北京市朝阳区惠新东街甲 4 号富盛大厦 2 座　邮编　100029）
（CITIC Publishing Group）
承 印 者：北京楠萍印刷有限公司

开　　本：787mm × 1092mm　1/16　　印　　张：27　　字　　数：400 千字
版　　次：2016 年 8 月第 1 版　　印　　次：2016 年 10 月第 2 次印刷
京权图字：01-2016-5558　　广告经营许可证：京朝工商广字第 8087 号
书　　号：ISBN 978-7-5086-6584-9
定　　价：79.00 元

资产证券化可能成为区块链最好的一个应用

/谢平

区块链，这个原本“高冷”的技术词汇，自2015年以来，引起了当前一波又一波最为火热的争议。到底什么是区块链呢？一般人都是因为知道比特币而知道了区块链，也都知道区块链是比特币的一项关键底层技术，通俗些说，它就像是一个数据库账本，安全记录所有的比特币交易信息。按照专家们更为专业的解释来说，该技术的实质是，不同的节点共同参与的分布式数据库，是一个开放式的公共账簿。从数据包形成区块，中间有一个加密的哈希值计算（密码学技术），把不同时间段的交易信息链接起来，就形成了区块链。

信用是金融活动的根基。具体到金融行业，人们正是希望能够通过区块链技术，低成本地解决金融活动中的信任问题。传统金融体系安排中，所有金融活动的监管及中介机构，包括产品登记、证券发行与交易、信息披露、资金托管等方面，都是解决信任问题或者说金融中最为核心的信息不对称问题。由于信任问题是一直难以解决的社会问题，所以，我们这个社会有很多的公信力机构。从反面来说，本次让市场投资者失去信心的长达10年的全球金融危机，全球货币与资产价值的不稳定，就是数字货币和区块链技术被国内外众多金融机构和个人追捧的一个重要背景，区块链技术给我们创造了一

个用“共信力”来解决公信力问题的途径。

互联网科技与传统金融机构有待进一步的融合。比如银行业，就需要更加重视业务经营管理的数字化、智能化建设，更加深入地推广应用移动互联、大数据、云计算、人工智能等先进技术，以科技改造业务、以科技推动创新。正是在这样的思想和认识下，我认为区块链技术也可能为包括银行、保险在内的机构提供当前许多问题的解决方案，不然当前很多以该技术为核心的金融科技公司并没有存在的必要。另一方面，银行家们也明白，区块链不会是银行终结的信号，区块链可以帮助银行和金融机构寻找新的机会，更好地服务客户。

由于对数字货币与区块链有一定的兴趣，希望增加认识和了解，我与本书译者林华教授畅谈了上面的认识与体会。这本著作是根据普林斯顿大学公开课改编的一部教材，主要讨论了比特币的一系列重要问题。比如，书中着重介绍了比特币的运作方式、比特币与众不同的技术知识、比特币安全性如何保障、比特币的匿名性特征、区块链如何帮助比特币实现没有身份的共识、人们在比特币这一平台上可以创建哪些应用程序、比特币的存储和使用、比特币挖矿、比特币监管，以及作者们对比特币的未来发展展望。我认为，在阅读完如此专业的教材之后，对当前热议的比特币和区块链的各种争议观点，我们就可以具备去伪存真的能力，或许还能掌握基础概念，并能够开发出安全的、能与比特币网络互动的软件，甚至能够把比特币相关理论应用于自己的项目中。

林教授曾经告诉我，资产证券化与区块链有一个很好的结合点，这是我非常感兴趣的一件事。众所周知，区块链被人们认识主要起源于比特币。比特币的本质是数字货币，区块链的本质在于它是一个分布式账本，而货币系统本身就是一个账本，这是它们能够天然结合在一起的很好解释。只不过，原来的货币系统账本是由央行控制和维护的，现在区块链则是分布式的（也有说成所谓的去中心化），是大家一起共同维护的一个账本。

资产证券化和区块链如何结合呢？一直专长于资产证券化的林教授告诉我说，数字货币的一个延伸在于代币（Token Coin），什么是代币呢？就是把资产

变成货币，代币作为资产使用权的证明，或者资产内在价值的所有权证明。资产变成货币，就是一种证券化。如果我们能够建立一个账本，将资产证券化池子中的资产，全部挪到这个账本上，基础资产的各种特征都做好标记，不断循环，按交易时间更新区块，不可篡改，定期跟踪，就能够实现资产证券化与区块链的一个有效结合。资产挪到账本，还需要从三个层面来说，第一个层面是资产，第三个层面是账本，中间需要一个开关或者说场景，形成一个映射关系，即将资产映射到对应账本上，实现所谓的货币化。中间层需要一个场景，最可能的场景就是交易所，可以实现资产和货币的交易。

当然，我和林华教授都一致认为，这背后还有一个担心，从金融角度来看，区块链在技术上仍然不够成熟，尤其是在交易环节。国内外许多专家都明确估计过，区块链技术可能还需要 3 ~ 5 年时间才能真正成熟。在这样的背景下，如果区块链技术被滥用，就会酝酿很大的风险，就好比前两年的比特币投机潮一样。我认为，如果区块链在交易活动中的跟踪、项目资金使用的全程监控以及智能资产合约所需要的风险控制措施等效用发挥不出来，只是利用分布式的分散管理效果，希望在没有一个第三方公信力机构的情形下保证信用，其结果会很容易搞得像 P2P 中的债权分拆分包一样，现实中的风险并没有缩小，只是被转移分散到广大投资人中去，最后出现我们看到的 P2P 机构跑路现象，更要防范的是，现有的 P2P 都将自己做的债权分拆，包装成所谓的区块链金融。正如北京金融工作局霍学文局长近期所说的，“如果现在不规范区块链技术，它又会成为非法金融活动的来源”。因此，我个人也强烈呼吁，我们要发展的其实是符合金融监管和行业规则的技术创新，如果在区块链技术基础上从事不规范的金融行为的话，也会造成新的非法集资或者金融不稳定的来源。

不过，我依然看好区块链技术在金融领域的运用，它不仅仅是货币创造，而且是价值传输与公共账户。现在国内外很多金融机构在价值传输，比如在支付结算、资产登记以及资产转让等方面也都有积极的探索。同时，由于区块链是一个公开、透明、可追溯、不可篡改的分布式总账系统，区块链技术可以有效降低支付、清算、结算步骤的出错率，同时监控每一步资金的流入流出情况，

是推动诚信社会建立的有效手段，区块链有利于金融监管的一面。随着监管与市场主体对区块链技术的认识不断加深，以及该技术不断走向成熟来保证资产的真实性，和林教授一样，我相信资产证券化极有可能成为区块链最好的一个应用。

基于以上的理解和认识，我欣然为林华教授这部翻译作品作序。

谢平

2016年7月

区块链到底是什么？

/肖风

有关于区块链是什么的话题，在时下的中国，可能已经被包括我自己在内的人说成了陈词滥调了。但是每每我们都会看到这样一种情形：一些我们认为已经是常识的概念，却往往别有洞天！借着《区块链：技术驱动金融》中文版出版之际，我愿意把我最近对区块链概念的反刍心得写下来，作为这本书的推荐序。

区块链首先是一种社会思潮。它预示着人类社会转型、换代的新时代的到来。区块链的社会学基础是凯文·凯利《失控》一书里观察及论述到的基于生物逻辑的自然、社会、技术的进化规律：分布式、去中心；从边缘到中心再到边缘，从失控到控制再到失控。微信之父张小龙奉《失控》为自己行动指南的行为，最好地说明了互联网时代的组织及经济发展规律已经变了。区块链的技术基础是分布式网络架构，正是因为分布式网络技术的成熟，去中心、弱中心、分中心及共享、共识、共担的组织架构、商业架构和社会架构才有可能有效建立起来。本书就是从工程技术的角度来介绍基于分布式网络架构的区块链技术的，分布式网络架构对人类社会的影响和冲击，也许我们都还无法估计，不可测量！

当然，任何事物都是精华与糟粕相生相伴、优点与缺点共存共荣的，区块

链技术也一样。在社会实践中我们已经看到，传统金融机构在接受区块链技术的精华的同时，已经扬弃了区块链技术当中的纯粹去中心化的无政府主义色彩和对人人都可以发行货币的去管制、去监管的追求。

区块链其次是一串技术组合。第一，它是分布式账本：全部机构一本总账、各种事务一本总账；第二，它是新型数据库：没有中心机房，没有运维人员，第三方按共识算法录入数据，非对称加密算法保证数据安全，数据客观可信，不可篡改；第三，它是智能合约：是一段能够自动执行约定条件的计算机程序，依靠智能合约技术，理想中的世界就好像一台精密运行的计算机，一切都可以事先约定，编成代码，依程序行事；第四，它是TCP/IP模型（互联网模型）里的点对点价值传输协议，它的发明标志着过去20几年，互联网技术在帮助人们更好地进行信息传输之后，开始帮助人们可以不借助任何第三方的信任背书，点对点、端到端、P2P地来传递、交易、支付、汇兑价值物。互联网从此进入新时代：价值互联网时代到来了！

区块链还是FinTech（金融科技）的核心。继互联网金融之后，金融科技最近大热大火。我们注意到，前几年互联网金融在中国活跃的时候，欧美国家几乎不为所动。而最近一年，欧美国家反过来把金融科技的火把传输到了中国，在互联网金融一地鸡毛的时候，点燃了中国金融创新的新热点。一开始我也认为，互联网金融和金融科技应该就是一回事。但细细想想，它们之间虽然没有本质不同，却还是重心各有侧偏。互联网金融侧重于场景革命，而金融科技侧重于技术革命；进一步，互联网企业拥有场景优势，所以在互联网金融阶段挟场景的优势，略胜传统金融机构一筹。其实就连互联网公司本身，也有场景能力的高低之分，电商和社交网络公司创建场景的能力最强，所以互联网金融的能力也就最大。其他类型的互联网公司，基本难以望其项背。

而就金融科技而言，侧重的是云计算、大数据、机器学习、人工智能等创新技术。技术是中立的，这意味着：一是技术公司固然有技术先发优势，但金融机构在应用先进技术方面也没有不可逾越的障碍；二是技术逻辑必须与业务逻辑结合才能创造价值，而金融机构在业务逻辑方面相比技术公司有优势，业务逻辑的经验积累也是需要时间和过程的。无怪乎最近我们看到太多的互联网

公司到金融机构挖人的消息，因为在金融科技阶段，互联网公司急需懂得业务逻辑的金融人才。

面对互联网公司的业务竞争，过去几年金融机构的应对举措大概分三类：一是无力回天，沦为通道；二是热情拥抱，全面对接；三是自建场景，创新模式。我们其实在多年以前就已经看到过飞信与微信演绎的故事了，它已经充分说明了切勿以己之短搏人之长的道理。

金融科技有可能是金融机构在与互联网公司业务竞争中的一次最好的机会，因为技术面前人人平等。

所以我们看到，这一次华尔街表现出来的热情超越硅谷。华尔街的金融机构都纷纷表白自己是一家科技公司或马上将成为一家科技公司。

区块链可以算得上是金融科技里的核心技术。因为区块链技术是金融业的底层技术革命。大家知道，现代银行业起源于意大利。之所以起源于意大利，一是意大利是欧洲最早开始海洋贸易的地区，复杂的、高风险的海洋贸易必然需要相配套的金融服务；二是意大利人发明了复式记账法，使得复杂的经济活动在会计上可计量。复式记账法几百年来一直没有重大的改进，区块链技术将是自复式记账法被发明以来，人类社会记账方法的第一次革命性改进。作为分布式账本技术，区块链必将给任何需要记账的行业带来降低成本、提高效率、创新业务、创新服务的机会。金融业因为其早已经数字化的特点，首当其冲，也必先蒙其利！

最后，希望本书的出版，能够从工程技术层面，推动区块链技术在中国的发展，推动相关应用的落地。祝《区块链：技术驱动金融》一纸风行！

肖风

2016 年 8 月

INTRODUCTION | 译者前言

这是一本关于比特币和区块链技术的专业著作，起源于业内所熟知的比特币和加密货币技术的普林斯顿网络公开课。以普林斯顿大学计算机科学助理教授阿尔文德·纳拉亚南（Arvind Narayanan）为首的专家，与我们分享了他们关于数字货币与区块链的权威研究成果和重要理论观点。

目前，国内对于比特币和区块链技术的热捧和争议，或者将其过度神秘化，或者将其贬斥得一无是处，还有很多别有用心的人不适当地鼓吹，正是反映出人们并没有真正搞清楚到底什么是比特币，它在技术层面到底是如何运作的。"他山之石，可以攻玉。"本书讨论的是比特币的一系列重要问题。比特币是如何运作的？它因何而与众不同？你的比特币安全吗？比特币用户如何匿名？我们可以在比特币这一平台上创建什么应用程序？加密数字货币可以被监管吗？创建一种新的数字货币将会带来什么样的变化？未来将会如何发展？本书中，作者承认比特币及区块链技术为各种领域带来了颠覆性的创新，但他们并不认可那种以去中心化为目的的观点。

我和王勇教授积极联系与申请，与中信出版社合作，通过激烈的竞争拿到了本书的翻译版权，书中内容的专业性非常强，从翻译初稿到终稿，经过接近一年的辛勤和努力，终于完成了本书的翻译。

在此，我首先要感谢中国投资公司前副总经理谢平先生和中国万向控股有限公司副董事长肖风先生不辞辛苦，亲自提笔为本书作推荐序。感谢金融时报社社长邢早忠先生、北京市金融工作局局长霍学文先生、浦发银行行长刘信义先生、厦门国家会计学院院长黄世忠先生以及深圳前海金融资产交易所总经理

唐斌先生为本书撰写推荐词，感谢您们的鼓励和支持。

我要感谢参与本书翻译的每一位译者。感谢帅初提供了1～9章的翻译初稿，蔡凯龙提供了其余两章和原版前言的翻译初稿。由于本书涉及多个专业领域，翻译初稿在专业性和体例统一等方面有待完善，我组织了所有译者进行重译和修订。其中，高晓婧负责前言与第7章，王勇负责第1章和第2章，洪浩负责第3章和第4章，蔡凯龙负责第5章，许余洁负责第6章和第10章，李耀光负责第8章、第9章和第11章。我、王勇和许余洁确定了全书的术语表，并一同再三审校全书。许余洁在整体校稿的基础上，还多次与出版社老师们对接书稿的最后内容的完善。每一位译者都在工作之余花了很多时间精推细敲、反复斟酌原文和译文，几经修订才使本书得以呈现在读者面前，感谢每一位译者的辛苦付出，也因此我们采用联合署名的译著方式。

另外，我要感谢杨昌丽、黄红华、韩世光、董方朋、王克祥等对本书在翻译过程中所提供的帮助。

最后，我还要感谢中信出版社编辑的精心编校，没有大家精益求精的团队努力与合作，这本书的中文版本不可能如此顺利与读者见面。

区块链技术在中国的健康发展，还是要基于我国监管的框架和逻辑下，与适当的行业进行有效结合。我们衷心地祝愿本书的引进，能够有助于大家正确理解比特币金融技术的创新与发展。

林华

2016年7月于北京

比特币和加密数字货币是当前的热门话题。乐观主义者认为比特币将从根本上改变人们的支付方式、全球经济甚至政治格局；悲观者则认为它生来就不完美，其失败是注定且彻底的。

究其根本，这些分歧之所以存在，是因为人们没弄清楚到底什么是比特币以及它是如何运作的。本书的目的就是帮助人们跳过噱头切入重点，看清比特币的特殊性。要真正了解比特币的特殊性，我们需要了解它在技术层面的运作模式。比特币是一项新兴技术。把它与现有技术进行简单类比，很难帮助我们做到这一点。

阅读本书需要具备计算机科学的基础知识，了解计算机的工作原理、数据结构和算法，拥有一定的编程经验。如果你是一名计算机专业本科生或研究生、软件工程师、创业者或技术爱好者，那么这本书很适合你。

本书将讨论比特币的一系列重要问题：比特币是如何运作的？它因何而与众不同？你的比特币安全吗？比特币用户如何匿名？我们可以在比特币这一平台上创建什么应用程序？加密数字货币可以被监管吗？创建一种新的数字货币将会带来什么样的变化？未来将会如何发展？

完成本书的学习之后，对比特币和加密数字货币的观点，你应该具备了去伪存真的能力；同时也掌握了基础概念，能够开发出安全的、能与比特币网络互动的软件；还可以把比特币相关理论应用于自己的项目中。

> 本书网上补充阅读材料中，还包含系列配套练习题，可以帮助你更深入理解每一章节。此外，你还需要运用到一些要求运用比特币的简化模型，来完

成一系列编程任务。本书的大部分内容都有视频，如有需要，可以在免费公开在线课程[①]上获得（补充材料获取网址为：http://press.princeton.edu/titles/10908.html）。同时，建议读者补充比特币相关知识，你可以阅读比特币维基、论坛、研究报告，并与比特币从业者及兴趣相同的人进行讨论。

① Coursera，是免费大型公开在线课程项目，由美国斯坦福大学两名计算机科学教授创办。旨在同世界顶尖大学合作，在线提供免费的网络公开课程。——译者注

通往比特币的漫长道路

/杰里米·克拉克（Jeremy Clarek）

在通往比特币的道路上，布满了无数失败的尝试。我收集了一份由约100个加密支付系统组成的名单。它们的技术基于电子现金（e-cash）和信用卡，在某些方面获得显著成就，见表0.1。其中一些是被广泛引用的学术研究成果，还有一些是已开发和测试过的实实在在的系统。在这份名单上，被大家所知的大概只有一个——贝宝（PayPal）。而贝宝之所以幸存，得益于它及时纠正了最初想在移动设备上进行加密支付这一想法。

这段历史会让我们吸取很多教训。比特币的想法从何而来？为什么一些技术成功了而另一些则一败涂地？如何成功地商业化那些复杂的技术创新？即便不去思考这些，它至少让我们明白，一个真实可行的基于互联网的支付体系是多么来之不易。

传统金融体系

设想在政府和货币出现之前，人们以物物交换的方式进行着交易。比如，爱丽丝（Alice）需要工具，鲍勃（Bob）需要药品。如果他们正好都有对方所需物品，就可以进行交换，满足各自所需。

但是，如果爱丽丝有食物，愿意拿食物换工具，鲍勃有工具但不需要食物，

表 0.1　一些优秀的电子支付系统和构想

ACC	CyberCents	IKP	MPTP	Proton
Agora	CyberCoin	IMB-MP	Net900	Redi-Charge
AIMP	CyberGold	InterCoin	NetBill	S/PAY
Allopass	DigiGold	Ipin	NetCard	Sandia Lab E-Cash
b-money	Digital Silk Road	Javien	NetCash	Secure Courier
BankNet	e-Comm	Karma	NetCheque	Semopo
Bitbit	E-Gold	LotteryTickets	NetFare	SET
Bitgold	Ecash	Lucre	No3rd	SET2Go
Bitpass	eCharge	MagicMoney	One Click Charge	SubScrip
C-SET	eCoin	Mandate	PayMe	Trivnet
CAFÉ	Edd	MicroMint	PayNet	TUB
Checkfree	eVend	Micromoney	PayPal	Twitpay
ClickandBuy	First Virtual	MilliCent	PaySafeCard	VeriFone
ClickShare	FSTC Electronic Check	Mini-Pay	PayTrust	VisaCash
CommerceNet	Geldkarte	Minitix	PayWord	Wallie
CommercePOINT	Globe Left	MobileMoney	Peppercoin	Way2Pay
CommerceSTAGE	Hashcash	Mojo	PhoneTicks	WorldPay
Cybank	HINDE	Mollie	Playspan	X-Pay
CyberCash	iBill	Mondex	Polling	

他想要药品。在这种情况下，爱丽丝和鲍勃就没法直接与对方交易。但是，如果有另一个人卡罗尔（Carol），他有药品，而且愿意拿药品换取食物。那么，这三个人就可以进行交易，各自获得所需物品。

当然，难点在于协调，即组织一群供需匹配的人在同一时间、同一地点进行交易。为解决这一难点，出现了两个体系：信用和现金。二者哪个更早出现，历史学家、人类学家和经济学家们就此争论不休，但这对本书的讨论无关紧要。

在上面的例子中，在信用体系里，爱丽丝和鲍勃可以与对方交易。鲍勃给爱丽丝她所需的工具，得到一个人情。换言之，爱丽丝欠下一笔债务，未来终将偿还给鲍勃。爱丽丝的物质需求即刻得到了满足，但她希望尽快还清债务，因此，她又有了新的需求。然后，爱丽丝又遇到了卡罗尔，她可以用自己的食物交换卡罗尔的药品，然后把药品给鲍勃。这样，她就偿还了债务。

对比而言，在现金体系里，爱丽丝可以购买鲍勃的工具，然后把食物卖给卡罗尔，卡罗尔再把药品卖给鲍勃，完成整个闭环交易。只要每场交易的买方有充足的现金，这些交易就可以按任意顺序发生。当然，最终的结果是，看上去现金似乎从未易过手。

很难说这两个体系哪个更优越。现金体系首先需要现金分配来触发，否则交易无法发生。信用体系不需要这样，但债权人需要承担债务人不偿还债务的风险。

现金还可以让我们知道物品的准确价值。物物交换时，我们很难说工具和药品到底哪个更值钱。现金交易把物品的价值标上数字，这就是为什么我们现在将这两种体系混合使用，即便使用信用，我们依然用现金来衡量所需偿还的债务金额。

这些观点被应用于许多场合，特别是用户在进行虚拟物品的线上交易时。例如，在进行点对点（peer-to-peer）的文件分享时，我们就可能遇到吃白食的人，他们只下载，不分享。进行文件交易可能是一个可行的解决方案，但是如何找到两个相互需要对方文件的人是个协调上的难题。在一些项目如莫佐（Mojo Nation）和学术构想如卡玛（Karma）中，用户自动获得一定数额的虚拟货币。接收文件时，用户可以用虚拟货币支付费用；向其他用户发送文件时，赚取虚拟货币。无论是接收还是发送文件，一个或者多个服务器跟踪记录用户的账户余额，而且可以把虚拟货币兑换成真实货币。虽然莫佐项目在推出货币兑换功能之前就消失了，但它算得上是我们现在使用的比特流（BitTorrent，一种内容分发协议）和塔荷（Tahoe-LAFS，一种分布式数据存储方式）的鼻祖。

网络信用卡的弊端

许多电子支付方式都可以根据信用和现金这两个基本概念进行分类。比特币显然属于现金类，但我们先来谈谈信用类。

信用卡交易是目前主要的线上支付方式。如果你在亚马逊这样的网站购过物，那么你应该很清楚流程。首先，输入你的信用卡信息，点击发送，亚马逊收到这些信息后反馈给“系统”，这一系统包括信息处理器、银行、信用卡公司及其他中介。

然而，如果使用贝宝交易，那么你体验的就是中介式结构风格。你和卖家之间存在一个中介公司，你把信用卡信息发送给中介公司，中介公司核准交易并通知卖家，并在每个交易日结束时与卖家统一结算。

这一结构的优势是，你不需要提供给卖家你的信用卡信息，规避了安全风险。你也无须向卖家提供个人信息，保护了个人隐私。劣势在于，它增加了复杂性，你和卖家无法进行直接交流，都得在中介公司开设账户。

如今，我们已经习惯在网络购物时提交个人信用卡信息，至少已经勉强地接受了这一点。我们也习惯了网络公司搜集我们的网络购物及浏览历史。但在20世纪90年代，网络尚是新兴事物，数据加密协议刚刚兴起，消费者对这些问题深感担忧，对网络购物的安全性并不信任。特别是，通过一个不可靠的渠道，把自己的信用卡信息提交给不知名的网络商家，这在当时看来，几乎难以置信。在这种背景下，中介式架构在当时引发了诸多兴趣。

1994年，第一虚拟公司（First Virtue）成立。它是一家较早成立的中介支付公司，也是最早设立完全虚拟化办公室的公司之一。顾名思义，它的员工遍布全国，通过互联网沟通。

第一虚拟公司的支付体系与贝宝现在的体系类似，只是早于后者很多年。用户注册，提交信用卡信息。当用户进行网络购物时，商家把详细的支付信息发送给第一虚拟公司，第一虚拟公司与用户确认支付信息，确认无误后批准支付。其中有两个细节值得注意：第一，所有沟通都通过电子邮件。那时网页浏览器刚刚开始全面支持HTTPS等加密协议，多方参与增加了加密该支付的复杂性（其他中介采用把信息嵌入URL链接或者在HTTP上定制加密协议的方式）。第二，用户有90天的拒付期，3个月之后卖家才能收到货款。现在，卖家可以立即收到货款，但是消费者依然可以索回货款或者对信用卡账单提出申诉。在这种情况下，商家必须把货款退还给信用卡公司。

20世纪90年代中期，出现了一个较有竞争力的中介体系，我们称之为安全电子交易协议（Secure Electronic Transaction，简称SET）。在SET体系中，用户无须把信用卡信息提供给商家，也无须在中介公司注册账户。进行网络购物时，用户的浏览器会将交易和信用卡信息加密存储在电脑上的应用程序里，只有中

介能够解密这些信息，甚至连商家都不能。这样，消费者可以放心地把加密过的信息发给商家。商家再把这些加密信息和它们自己的交易信息一同转发给中介。中介解密你的信息，与商家的交易信息进行对比，只有在双方信息一致的情况下，中介才会批准支付。

SET 由维萨（VISA）、万事达以及多家当时重量级的科技公司开发而成，包括网景（Netscape）、IBM、微软、威瑞信（Verisign）和 RSA（美国知名信息安全、加密数据公司）。它融合了多个方案，成为一个标准性体系。

一家叫网络现金（CyberCash）的公司采用了 SET 体系。这家公司在很多方面都很有趣。它们的产品除了处理信用卡支付交易之外，还包括一种叫作网络币（CyberCoin）的数字货币。这是一种小额支付系统，用于支付小额消费行为，比如，用户可以用网络币支付几美分在线阅读报纸的费用。这也就意味着，用户账户里的网络币余额一般都在 10 美元以内。但是，有趣的是，它们却能得到美国政府联邦存款保险公司（FDIC）对每个账户高达 10 万美元的投保金额。

更有趣的是，网络现金公司运行时，美国政府限制加密技术的出口，因为当时加密技术被认为是一种武器。当然，现在这种限制已被废除。但在当时，这也就意味着国外用户无权下载包含加密技术的软件。但是，网络现金公司得到美国政府的特批，国外用户可以下载它的软件。政府给出的解释是，从网络现金公司的软件中提取加密技术远比从头开发一套全新加密技术要难得多。

最后，许多人怀疑网络现金公司与其他为数不多的几家公司一起受到千禧虫感染（Y2K bug），向部分客户重复收费。2001 年，网络现金公司破产，其知识产权被威瑞信收购，接着转卖给贝宝，贝宝屹立至今。

为什么 SET 体系行之有效？根本原因在于它的认证机制。认证就是把加密过的身份，即公钥（public key），与现实身份连接起来。网站要从像威瑞信这样的认证授权公司获得认证，用户的浏览器才会判定它是安全的（通常会显示一个锁形状的图标）。网络现金公司和 SET 体系认为，安全性比操作的便捷性更重要，因此，它们不仅要求服务商和商家，还要求客户也必须获得认证。获得认证的过程类似于报税一样烦琐，因此，这个系统简直是场灾难。几十年来，大多数用户都拒绝使用要求终端客户认证的系统，这种系统只会出现在学术论文

里。比特币巧妙地避开了这一难题，而且无须用户的真实身份。比特币系统通过公钥本身来辨别用户身份。我们将在第 1 章探讨这个问题。

20 世纪 90 年代中期，正当 SET 体系标准化时，万维网联盟（World Wide Web Consortium）也在探索如何将金融支付方式标准化。它们试图扩展 HTTP 协议，这样，用户不需要其他软件，通过浏览器就可以完成交易。事实上，它们对如何扩展 HTTP 协议提出了一个总体方案，也给出了一个用户支付案例。但这从未付诸实践，整个扩展框架并未应用于浏览器中。近 20 年后，2015 年，联盟宣布重新考虑扩展计划，这次，比特币将成为该标准化进程的一部分。但是考虑到以往的失败教训，我对此并不乐观。

从信用到（加密）现金

现在该谈谈现金体系了。如前所述，将现金和信用进行比较，我们发现，现金体系需要启动自循环，但优势在于，它规避了买家拒不偿还债务的风险。此外，现金体系还有另外两个优势：第一，更好地确保了用户的匿名性。信用卡与个人信息绑定，因此，银行可以追查消费者的所有消费记录。但是，如果使用现金交易，就与银行无关，卖家也无须知道消费者的个人信息。第二，现金支持线下交易，无须致电第三方获得交易批准。也许交易完成后需要把钱存入银行，但这要容易得多。

比特币没有这两个特点，但具备两个类似的功能。它的匿名性比不上现金。用户在使用比特币支付时，无须使用自己的真实身份，但是，如果用户不够小心谨慎，可以借助公开的交易账目和聪明的算法查出用户的交易记录并最终查出用户身份。我们将在第 6 章展开这个复杂又有趣的比特币匿名性问题。

比特币并不完全支持线下交易。但优点是，它不需要中央处理器，而是依赖点对点网络，这种网络跟互联网一样具有很强的修复能力。我们将在第 3 章讨论“绿色地址”和小额支付工具，它们可以帮助我们在特定条件和特定情境下进行线下支付。

大卫·乔姆（David Chaum）在 1983 年最早提出把加密技术运用于现金上的想法。我们可以拿现实中的例子来帮助理解。比如，我向人们发放纸条，上

面写着“拿到此条的人可以来我这里领取 1 美元”。假设人们信任我不会食言而且我的签名不可伪造，他们就可以像银行汇票一样流通纸条。事实上，银行汇票最初就是商业银行发放的支付承诺。只是到了近代，政府才开始介入，集中货币供给，用法律手段强制要求银行兑现票据。

我可以通过数字签名发放电子纸条，但那样的话，又会遇到一个“双重支付”（double spending）这一恼人的难题：收到表示一定金额的虚拟货币的数据时，人们可以复制该数据，然后传输给他人。假设人们的复制技术足够优秀，我们难以辨别哪些是初始数据，哪些是复制品，那么，我们能够解决“双重支付”的问题吗？

可能的解决方案是：我在发出的每份纸条上印上一串独特的序列号。当别人把纸条给你时，你检查一下我的签名，然后打电话给我，告诉我相应的序列号，询问印有这个序列号的纸条是否已被使用过。如果我告诉你没有，那你就可以放心地收下这个纸条。我会在账本上记录该纸条已被使用。你要做的是定期把收到的纸条交给我，我会再给你相同数量的印有新序列号的纸条。

这个方法是可行的。它在现实中施行起来颇为烦琐，但在网络上却比较简单明了，只要我设置一台服务器，用它来完成签名和序列号的记录工作。唯一的问题在于，因为难以匿名，它很难称得上是真正的现金。不管是发行还是兑现纸条，我都可以把序列号和个人信息一同记录在案。这也就意味着，我能够追踪你的所有消费行为。

乔姆提供了一个创造性解决方案。它不仅能够保护用户的匿名性，同时还杜绝了“双重支付”。它的方法是：我给你一张纸条，你把它的序列号记录下来，并且不要让我看见。然后我再签名，并不需要知道它的序列号。这在密码学里被称为“盲签”（blind signature）。选取一个较长、随机的序列号能够更好地保护你的利益，因为这样的序列号更有可能是独一无二的。我不必担心你选取一个使用过的序列号，因为这样你只会得到一个无效货币而遭受损失。

这是第一个真正意义上的电子货币方案。它虽然有效，但必须要有一个大家信任的中心机构管理运行的服务器。不仅如此，这个服务器还必须参与每笔交易。如果服务器停止工作，交易就不得不暂停。数年之后的 1988 年，乔姆与

两位密码学专家阿莫斯·菲亚特（Amos Fiat）和摩尼·纳欧尔（Moni Naor）合作，提出线下电子货币的概念。乍看上去这似乎是不可行的：如果用户把同一个电子货币支付给两家没有连入同一个网络或与同一家中心机构合作的不同商家，它们怎么能够发现并阻止这种行为呢？

与其去预防双重支付，不如关注事后当商家重新连上银行服务器的时候如何察觉。这才是比较聪明的做法。乘坐没有网络连接的飞机时，如果你用信用卡消费，真正的转账是在航空公司重新连上网络之后才发生的。如果你的信用卡被拒付，你会欠航空公司（或你的银行）一笔钱。仔细想想，传统金融体系的很大一部分就建立在如何检测错误和损失这一基础之上，然后才是收回损失或惩罚失误方。如果你给某人开一张个人支票，他不会知道这笔钱是否真实存在于你的账户里，但当他去银行兑现时被银行拒绝，他会追究你的责任。类似地，如果线下电子货币系统被广泛应用，国家应该制定相关法律，规定双重支付属于犯罪行为。

为了检测出双重支付，乔姆、菲亚特和纳欧尔三人提出了一种复杂的加密机制。简而言之，这套机制可以达到以下目的：发行方在电子货币中以加密方式嵌入你的个人信息，除了你本人，包括银行在内的任何人都无法解密。你用电子货币消费时，接收方会随机挑选一部分密码要求你解密，并将之记录下来。这种解密的内容不足以暴露你的身份。如果你用同一份电子货币双重支付，当两个接收方都去银行兑现时，银行可以把两份解密的信息合在一起，最终几乎可以肯定知道你的身份。

你可能会担心，万一有人陷害我双重支付呢？比如，你支付给我一份电子货币，我不去银行兑现成有我身份加密的新数字货币，而是直接拿着你给的货币进行重复消费。不必担心，这是行不通的，因为我在用它支付时，接收方会要求我解密一段密码，这段密码与之前你解密的那段密码肯定是不一样的，因此，我无法完成这一解密任务。

多年以来，许多密码学家一直在研究并完善这一机制。在乔姆、菲亚特和纳欧尔提出的构想中，假设一枚电子货币价值 100 美元，如果你想买一个价格为 75 美元的物品，你没法把这枚货币分割成 75 美元和 25 美元。你只能去银行，

把价值 100 美元的货币兑换成现金，再拿现金换取价值 75 美元和 25 美元的货币。但是，在一篇论文里，Tatsuaki Okamoto 和 KazuoOhta 用梅克尔树（Merkle trees）建立了一个可以分割电子货币的系统。梅克尔树在比特币里还会出现，我们将在第 1 章遇到它。这个机制的效率还有很大的提高空间。特别是，这一机制采用了由史蒂芬·布兰德斯（Stefan Brands）在 20 世纪 90 年代，詹·卡姆实（Jan Camenisch）、苏珊·洪博格（Susan Hohenberger）、安娜·莉斯卡亚（Anna Lysyanskaya）在 2005 提出的“零知识验证”（zero-knowledge proofs），带来了很好的效果。在第 6 章，我们将看到，零知识验证也同样被运用于比特币体系中。

继续回到乔姆。为把自己的想法商业化，他于 1989 年创立数字现金公司（DigiCash），应该是第一家致力于解决线上支付问题的公司。数字现金公司比我们之前提过的第一虚拟公司和网络现金公司早了整整 5 年。数字现金系统使用的现金叫电子现金，另外，它们还有一个名为“网络资金”（cyberbucks）的系统。包括美国的几家银行和芬兰至少一家银行在内的数家银行，确实使用了这个系统。这可是远在比特币出现之前的 20 世纪 90 年代，可能会让一些比特币推崇者大吃一惊，因为他们认为银行是惧怕科技、抵制创新的庞然大物。

当你需要交易时，你点击一条由资金接收方发回的链接，跳转至数字现金网页，同时，会开通一条反向链接连回你的电脑。也就是说，你的电脑必须能够接收外部链接，就像一台服务器。你需要拥有自己的 IP 地址，你的网络服务提供商也必须允许外部连接。如果连接成功，电子现金软件会在你的电脑上运用，然后你再批准交易，进行付款。

乔姆的数字现金技术获得了几项专利，特别是它使用的盲签技术。外界对他的行为是有争议的，因为专利妨碍了其他人用该技术进一步开发电子现金系统。但是几位经常在一个叫“网络朋克”（cyberpunks）的邮件组里互动的密码学专家则另辟蹊径。著名的中本聪（Satoshi Nakamoto）第一次向全世界宣布比特币系统就是在一个邮件组里，它的前身就是网络朋克，这绝非巧合。我们将在第 7 章探讨网络朋克运动及比特币的起源。

网络朋克的几位密码学家开发出了一种名叫魔法货币（Magic Money）的类

似于电子现金的产品。魔法货币虽然侵犯了电子现金的专利，但因为他们宣称它只用于实验目的，因此并未被禁止。魔法货币是一个很有趣的软件。它采用纯文本界面，你可以通过电子邮件发送交易信息，只需要把交易信息复制粘贴到电子邮件并发送给其他用户就可以了。当然，你需要使用 PGP（Pretty Good Privacy，一种加密软件）等终端对终端的电子邮件加密软件，以确保信息在传输过程中的安全。

随后，本·劳里（Ben Laurie）在其他人的帮助下创立 Lucre 系统。该系统试图用一种非专利技术替代电子现金中的盲签，其他则与电子现金系统大致类似。

另外一个由伊恩·戈德堡（Ian Goldberg）提出的方案则试图解决无法分割电子货币换取零钱的问题。他的思路是：当你没有零钱而向商家支付了过多金额时，如果商家有货币，它会转回给你超额支付的部分。但是应该注意到，这一想法带来匿名性问题。如前所述，在电子现金系统里，付款人匿名而商家不匿名。但是当商家找零时，商家实际上成了付款人，因此他们是匿名的。从你收到商家的找零之后，你需要去银行兑现，这时，你又是不匿名的。这一系统无法确保用户的匿名性，因此，伊恩·戈德堡又重新设立了一个系统，在这个系统中有不同类型的货币，能够确保用户在匿名的情况下既能消费又能收到找零。

为什么数字现金最终失败了呢？主要原因在于它没能说服银行和商家使用它。因为使用这一系统的商家不多，用户也就不愿意用它。更糟的是，它并不或没有支持好用户和用户之间的交易，只侧重于用户和商家之间的交易。因此，商家不接受它，这个系统就很难激发其他人的兴趣。最终，数字现金败给了信用卡公司。

另外，比特币既支持用户和商家之间的交易，也支持用户和用户之间的交易。事实上，比特币体系并不把用户和商家区别开来。比特币的成功很大部分大概要归功于它对用户－用户间交易的支持。从一开始，每位比特币用户都可以发给其他用户，因而整个比特币社区都努力争取人们对比特币的支持，并促使商家也接受它。

数字现金公司的最后几年，它试图通过防侵入硬件来预防双重支付，不再把重心放在双重支付发生后的检测上。在这套系统里，有一种叫作钱包或者类似于卡片的设备。这个设备会记录你的账户余额。消费之后，余额减少；充值之后，余额增加。这个设备的用处是，没有人能够更改计数器数额，不管是通过物理手段还是电子技术。因此，当计数器归零时，倘若没有继续充值，用户都无法消费。

许多公司推出过带有防侵入硬件的电子现金系统。数字现金后来与一家叫CAFE的欧洲公司合作。另一家叫Mondex[①]的电子钱包公司也是基于这个想法创立的，后来被万事达收购。维萨（Visa）也有类似的系统，名为维萨货币（VisaCash）。

在使用电子钱包时，使用者既持有一张智能卡片，又拥有一个“读卡器”（wallet unit），两者均可进行充值。使用者之间直接可以互相进行支付。支付方将智能卡插入读卡器中，钱即转入读卡器。接受方将卡插入读卡器，钱就转入第二次插入的卡里。这一过程交换的是数字货币，是匿名的交换流程。

Mondex公司在几个地方推广其技术，其中一个城市正好离我的家乡安大略省圭尔夫市不远。你大概已经猜到，这项技术并没有被广泛使用。主要原因是，电子Mondex卡片跟现金类似，一旦丢失或者被偷，钱也就丢了。更有甚者，如果卡片发生故障，或者读卡器无法读卡，就没法知道卡里余额是多少。这种事情真正发生时，Mondex公司一般会自担损失。它们会假定卡里有余额并赔偿用户损失，这自然是一笔不小的开支。

此外，这个系统里的钱包反应比较慢。用信用卡或现金支付要快得多。商家都不喜欢拥有太多支付终端，对它们来说，一个信用卡POS机就够了。多重原因加在一起导致了Mondex公司的失败。

尽管如此，Mondex公司的用户卡是有小芯片的智能卡片，这项技术事实证明是相当成功的。如今，在很多国家，包括我所在的加拿大，每张信用卡和借

① Mondex是一种灵活的电子现金，是当今世界上几种主流的开放式通用电子钱包标准。最初是英国西敏寺银行开发的电子钱包，是世界上最早的电子钱包系统。——译者注。

记卡都采用了智能卡片技术。它们的目的不是防止双重支付。非现金技术中不会存在双重支付的问题，因为银行而不是卡片记录你的账户余额和可用信用。智能卡片的目的是用于认证，也就是说，它为了证明你知道自己账户的PIN。虽然用途不同，早在银行广泛采用该技术之前，Mondex公司就已经开始运用这项技术。

凭空发行货币

如果你有一个价值100美元的电子现金，那怎么能够保证它的确价值100美元呢？数字现金给出的答案很简单：要想获得一个价值100美元的电子现金，你必须从你的银行账户取现100美元，交给发行电子现金的银行。但要实现这个目的可以通过不同的方式，不同的公司采取的方法也各不相同。设想一个小概率事件：如果一个政府授权某家银行发行电子货币，凭空创造新电子现金，会怎么样呢？网格现金（NetCash）就是基于这一假设创立的，但是它并未真正实施过。电子黄金（E-Gold）则采用一套完全不同的体系，它在保险库中存入一定量的黄金，根据黄金价格发行电子货币。一家名为数字黄金（Digigold）的公司并不完全依赖黄金，但也有部分黄金储备。

归根结底，这些方式都是使电子货币的价值随美元或某种特定商品的价值而浮动。如果美元价值上升或下降，你的电子货币价值就相应地上升或下降。另一种比较激进的方案，就是使电子货币自成体系，其他货币不会影响其发行和价值。

要想创造一种自由浮动并且具有真实价值的虚拟货币，必须要设计出某种具有稀缺性的东西。其实，正是因为黄金和钻石的稀缺性，它们才会成为货币的储备。在虚拟世界，你可以这样设计你的系统，即虚拟货币只有在需要花一段时间解决了一定的数学计算（或“谜题”）之后方可生成，这样就保证了稀缺性。比特币体系中的“挖矿”就是这样的，我们会在第5章详细探讨。

通过解决数学计算来赋予虚拟货币价值，这一想法并不新鲜。早在1992年，密码学家辛提亚·沃克（Cynthia Dwork）和摩尼·纳欧尔（Moni Naor）首次提出这种方案，用来降低垃圾邮件问题。设想你每次发送邮件时，计算机都

不得不花几秒钟的时间解决一道数学计算题目。如果你没能附上题目的答案，收件人的邮箱会自动忽略你发来的邮件。对于普通用户，因为他们发送邮件的频率不高，不会带来太大麻烦。但对于想同时发送成千上万垃圾邮件的人来说，解决大量的数学计算几乎是不可能的。1997 年，亚当·贝克（Adam Back）在一个名为哈希现金（Hashcash）的体系中采用过类似设计。

要想阻止垃圾邮件，这些数学计算必须具备一定的特性。第一，垃圾邮件发送者解出一道题目之后，不能把这个答案附在他发送的其他邮件上。为了做到这一点，每封邮件会对应一个数学计算题目，题目内容取决于发件人、收件人、邮件内容和发送时间。第二，收件人无须重复解题的烦琐过程，就可以轻松地检查发件人附上的答案。第三，题目之间应是相互独立的，也就是说，解决一道题目不会减少解决其他题目所需的时间。第四，随着硬件性能的提升，解决数学计算变得越来越快、越来越容易，收件人必须要调整他们收到的答案的难度。通过密码学中的哈希方程（hash functions）设计的题目可以满足以上要求，我们将在第 1 章学习它。

比特币使用的数学计算与哈希现金的基本类似，只是进行了微小的改进。比特币能做的比哈希现金多得多，毕竟，要解释比特币需要一整本书呢！我之所以提这些，是因为哈希现金的创始人亚当·贝克曾经说过："比特币只是把哈希现金进行通货膨胀控制得到的延伸产品罢了。"我觉得这话有点过分了，就像说："特斯拉只是在轮子上加上电池而已。"

正如密码学里任何一个优秀的想法一样，数学计算题目有许多变体，每个变体具有些微不同的特性。其中一个构想来自维莱特（Rivest）和夏马尔（Shamir），他们提出了 RSA 加密系统（RSA 中的 R 和 S 分别为 Rivest 和 Shamir 的首字母）。研究哈希现金之后我们发现，解决一系列数学计算题目的成本就是解决单个题目的简单叠加。但政府发行货币时，成本可不是这么计算的。单是纸币上的防伪技术，政府就需要投入巨大的初始成本来购买设备，施加安全措施等。但是一旦研发出了防伪技术之后，成本就会降低，印一张货币和印一百张的成本差别并不大。换言之，发行纸币的固定成本很高，但浮动成本很低。维莱特和夏马尔想要设计的数学计算题目具有类似的成本结构，这样，发行第

一个电子货币需要巨大的计算量，但接下来就会变得很简单。他们的设计也运用了哈希方程，但使用方式不同。我们不打算讨论他们的详细方案，但他们要解决的问题是非常有趣的。

人们为什么没有广泛使用哈希现金来阻止垃圾邮件呢？也许是因为垃圾邮件问题还没有足够严峻。对大多数人来说，垃圾邮件只是个恼人的小问题，并没有严重到他们愿意用计算机算力来解决它。现在，我们有了垃圾邮件过滤器，能够有效地阻挡垃圾邮件。另外一个可能的原因是，哈希现金无法真正阻止垃圾邮件。特别是，现在大多数垃圾邮件发送人通过僵尸网络，用病毒大量入侵他人电脑，批量发送垃圾邮件。他们也可以通过这些电脑来获取哈希现金。所以，通过数学计算进行限制的想法还在不断发展中。在一些替代网络协议的构想中，如小型 LT 协议（MinimalLT），我们还可以看到这一思路。

把一切信息都记录在数据库账本中

区块链是比特币的另一项关键技术，它像一个数据库账本，安全记录所有的比特币交易信息。区块链的理论基础由来已久，可以追溯到哈勃（Haber）和斯托尔内塔（Stornetta）在 1991 年开始发表的一系列论文。他们提出的不是虚拟货币体系，而是一种可以安全地对数字文件进行时间戳记录的方法。时间戳是为了记录文件创建的大概时间。更重要的是，时间戳可以准确反映文件创建的先后顺序：如果一份文件比另一份文件更早创建，可以从时间戳中看出来。时间戳的安全性体现在文件的时间戳一旦生成，无法更改。

用户发送文件时，哈勃和斯托尔内塔设计的体系能够向客户提供时间戳服务。服务器收到文件时，它会用当时时间和指向之前文章的链接或者指针作为签名，来签名该文件并产生包含签名信息的认证，见图 0.1。这里所说的指针，指向的不是一个具体地址，而是一串数据。也就是说，如果该数据被更改了，那么这个指针也就自动失效。在第 1 章，我们将学习如何使用哈希方程来创建这种指针。

这种协议实现的效果是：每份文件的认证都确保了上一份文件内容的完整性。其实，反复运用这一理论：每次认证基本上都保障了这个认证点之前的所

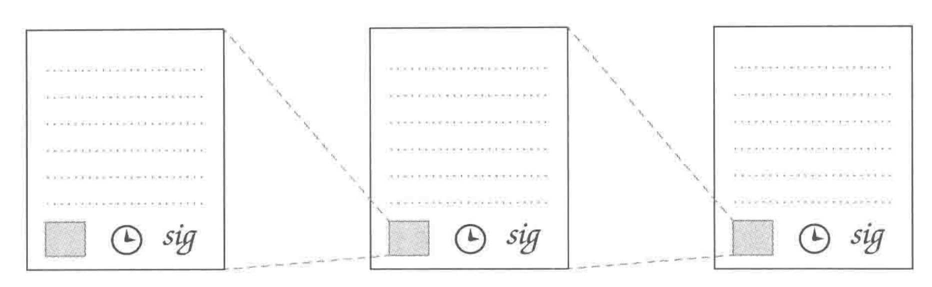


图 0.1　链接的时间戳

注：要想对一份文件进行认证，时间戳服务器必须包括指向之前文件认证的哈希指针，当前时间和文件内容本身，并用这三条信息来对文件进行签名。

有文件和认证的完整性。假设这个系统中的每个用户都能记录包括自己的文件、之前和之后的文件的认证在内的几个认证信息，那么合起来，就可以确保整个文件系统不会被更改。特别是，文件的先后顺序被保存了下来。

随后的一篇论文提出了一个可以提升效率的方案：不必单独链接各个文件，而是把它们集合成块，然后在一条链中链接整个块。在每个块里，文件通过树状结构而非线性结构的方式相互链接。这一方法减少了在整个系统中查找特定文件所需的工作量。图 0.2 展示了这一混合而成的体系的工作方法。

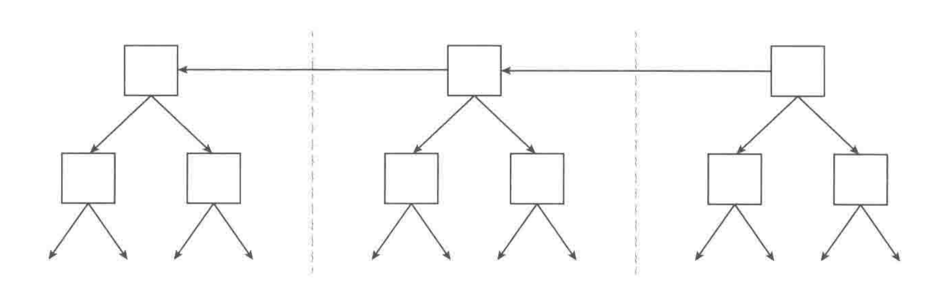


图 0.2　高效的链接时间戳

注：箭头表示哈希指针，点状垂直线表示时间间隔。

这一数据结构形成了比特币区块链的框架，我们在第 3 章可以看到这一点。比特币将它进行了微妙但至关重要的改进，它用一种类似于哈希现金的协议来降低在区块链里增添块的速度。这种改进对比特币的安全性带来了深远而有益的影响。比特币体系通过一群未被认证过的节点，即“矿工”，来记录事件，不再需要认证过的服务器。通过每个矿工而不是普通用户记录块的信息。任何人

都可以通过解决数学计算，新建块，而成为一名矿工。比特币还舍弃了签名，只依靠哈希指针来确保数据结构的完整性。最后，真正的时间戳对比特币来说不是很重要，整个系统的意义在于根据先后顺序记录交易信息，并确保它不能被篡改。事实上，比特币块并不按固定时间表产生。在比特币系统里，平均每10分钟产生一个新的块，单相邻的两个块的时间间隔会有较大的差别。

比特币从根本上融合了用数学计算来控制新币的产生和用安全的时间戳来记录交易信息并防止双重支付这两种思路。在比特币之前，有人曾提出过，不这么成熟但也融合了这两种思路的方案。比如，戴伟（Wei Dai）在1998年提出的B币（b-money），任何人都可以通过一个类似于哈希现金的系统创造虚拟货币。它跟比特币类似，也有一个点对点网络。每个节点维护一个数据库账本，但它不同于比特币的区块链，并不记录全部交易信息。每一个节点都有它自认为是准确的记录每个人账户余额的账本。

尼克·萨博（Nick Szabo）还提出一个类似的方案，名为比特黄金（Bitgold）。萨博宣称他早在1998年就有了创建比特黄金的想法，但直到2005年才在博客上公开发布。我之所以提这件事，是因为《纽约时报》记者纳萨尼尔·波普（Nathaniel Popper）曾写过一本关于比特币历史的不错的书，他发现，公开发布比特黄金的那篇博文的发表时间被修改过，改成了中本聪正式发布比特币之后的两个月。他跟许多观察者一样，认为萨博就是中本聪，即使萨博否认。他认为，萨博修改博文发表时间就是证据，这样，后者就可以掩盖自己在比特币发布之前就已经发明了比特黄金这一事实。

但这一证据并不可信。仔细阅读博文，你就会发现，萨博明确表示自己在1998年就有了比特黄金的想法。他没有试图更改这些时间。更为合理的解释是，比特币开始流行后，他把那篇博文置顶，这样，人们就可以看到他在比特币之前就有了类似的构想。

比特币与B币和比特黄金有很多重要的不同之处。B币和比特黄金通过数学计算直接创造货币。任何人都可以解题，答案本身就是货币。但在比特币体系中，解决数学计算并不构成货币，只是确保区块链安全，间接地在有限时间里创造新货币。此外，B币和比特黄金靠时间戳给货币的创造和转账签名。但

比特币不需要被认证过的时间戳，它只是用时间戳来保存区块和交易的先后顺序。

最后，如果服务器和节点对数据库账本的记录不符，B币和比特黄金并没有提供一个明确的解决方案。两位发明人所著文章暗示的解决方案是，由大多数人来决定到底哪个是对的。但是，因为任何人都可以用不同身份设置一个或一百个节点，这个解决方案并不可靠，除非有一个管理员来监管网络入口。比特币则恰恰相反，如果攻击者想更改数据，他必须要比其他所有人加起来的解决数学计算的速度还要快。这样就保证了安全性，还可以让我们量化直观地看到整个系统有多安全。

B币和比特黄金都不是正式发布的体系，B币是在一篇发表在邮件组的文章中提出的，比特黄金则是在几篇博文中提出的。二者未被真正实施和广泛应用。不像比特币白皮书，它们也没有给出详细设定或程序源代码。二者都涉及可能无解的若干问题。其一是前文讨论过的数据库账本不一致的问题。另外的问题是，如何设置创造新货币的数学运算的难度？具有同等运算能力的硬件随着时间的推移越来越便宜，针对这一问题，比特币采用了周期性自动调节运算难度的机制。B币和比特黄金没有这样的机制，因此，它们的货币会因为创造难度降低而贬值。

关于中本聪的猜测

你大概知道，中本聪是比特币创造者的化名。他的真实身份依然是一个谜团，但在比特币早期，他还是比较活跃的。我们可以从他留下的这些印记中，探讨一下他本人，比如说他是从什么时候开始研究比特币的？我们谈论过的那些早期构想对他有什么影响？是什么在激励着他？

中本聪说他从2007年5月左右开始编写比特币。他选择匿名并不表示他会在这一点事情上说谎，因此，我姑且相信他所说的。他于2008年8月注册域名bitcoin. org。同时，他开始给一些他认为可能会对比特币感兴趣的人发送邮件，阐述他的想法。2008年10月，他公开发表白皮书，解释比特币协议。此后不久，他又公开了比特币的源代码。随后两年，他在论坛上发布信息，与许多人

写邮件交流，回答人们的关切。在编程方面，他对源代码进行了多次修改。他和几位开发人员一同维护源代码，修复补丁。2010 年 11 月，别人逐渐开始接手比特币项目，而他却不再出现。

我用“他”来表示中本聪，但其实我并不知道中本聪到底是男性还是女性，只是因为中本聪是个男性名字而已。此外，我认为中本聪是一个人，而非一个团体。原因是，仔细研究中本聪所有的网络交流记录，在两年的时间里，一个团队里的多个人共用一个账号回复邮件，修改代码，保持风格、语气和内容一致，这简直难以想象。更为合理的解释是，展现在我们面前的中本聪的所有行为是由同一个人完成的。

此外，从他的文章和所打的补丁来看，这个人完全了解比特币的整套代码体系和设计细节。我们有理由相信源代码和白皮书由同一个人所写。最后，可能在一开始有人帮助过中本聪。虽然如此，比特币正式发布之后，我们可以看到，中本聪在得到帮助之后，会很快对其他有贡献的帮助者表示感谢。从这点性格来看，他应该不会在接受别人的帮助之后闭口不言，故意误导人们相信比特币是他一个人的作品。

我们可能会想：“中本聪知道电子现金的历史吗?”为更好地弄清楚这个问题，我们可以看看他在比特币网站上早期发布的白皮书里的引用和索引。在白皮书中，他引用了一些基本密码学和概率论的论文，也引用了我们之前所讨论的时间戳。因为比特币里的区块链与他所引用的内容相似度极高，自然而然地，我们会认为他有参考别人的区块链设计。他还引用了哈希现金，它的数学计算与比特币的非常类似。此外，他还引用了 B 币。随后，他又在网站上添加了比特黄金和一个由哈尔·芬尼（Hal Finney）设计的重复利用计算数学方案的参考索引。

但是，从与中本聪有早期交流的人公开的邮件来看，我们发现，B 币是在亚当·贝克的提议下才加入比特币体系的。随后，中本聪给 B 币的创造者戴伟发邮件。从邮件中可以看出，是戴伟告诉了他比特黄金。因此，激发中本聪创造比特币的，并不一定是这些方案。他之后与哈尔·芬尼有过多封往来邮件，这可以解释他为什么在网站上或者其他地方引用了芬尼的成果。

基于上述信息，比较可信的推断是，在创建比特币体系时，中本聪只知道电子现金、哈希现金和时间戳，认为只有这些与比特币是相关的。然而，等他知道B币和比特黄金的时候，他才发现，这两个也与比特币有很大关系。2010年，维基百科主编认为比特币不值一提，准备删除比特币词条。中本聪跟另外一些人讨论如何编写比特币词条，好让维基百科接受它。中本聪建议这样描述比特币："比特币是戴伟在1998年在网络朋克中所提到的B币构想和尼克·萨博提出的比特黄金的具体实现。"可见，中本聪这时确实把比特币看成二者的延伸或具体实施，以便更好地解释比特币的工作原理。

那么，中本聪创建比特币时，他知道其他体系吗？比如我们提过的乔姆的电子现金和信用卡方案。这个很难讲。我们找不到他了解它们的证据，但也有可能他虽然知道，但并未提及它们，因为它们与比特币无关。比特币采用了完全不同的去中心模式，没有理由去提已经失败了的中心化体系。

中本聪自己也表明了这一点。他在发表在比特币论坛里的一篇文章里，曾粗略地提了一下乔姆的电子现金体系。他当时正在写一篇关于opencoin. org的文章，他说他们似乎在"讨论老一套的乔姆中心造币体系，但也许是因为他们别无选择，也许他们会对新的方向感兴趣。20世纪90年代以来所有的虚拟货币公司全都失败了，这导致许多人对这一行业非常不看好。我希望，人们可以看到，这些系统之所以失败，显然是因为它们的中心化控制这一特性。我想我们正在首次尝试建立一个去中心化的非认证系统。"从这段话里，我们可以清晰地了解到中本聪是怎么看待之前的系统的，特别是，他认为比特币与它们是不同的。去中心化这一特性确实真正让比特币从其他虚拟货币脱颖而出。

中本聪写下的另一段话暗示他不是学术派人士。大多数学术研究者先有了构思，然后写下来，再把自己的构思付诸实施。中本聪说他的方式截然相反："我在建造比特币时，其实是倒着来的。我必须写下所有代码，才能使自己相信我是可以解决任何问题的，然后我才写下理论。我认为我能在写出具体设计细节之前就可以公开代码。"

中本聪是一个神秘的人，值得一提的是，跟所有人一样，他也会犯错，也无法预测未来。比特币的源代码和设计中都存在很多漏洞和瑕疵。例如，比特

币体系有一个可以向 IP 地址发送比特币的功能。虽然当时人们并未发现，但现在看来，这一设计十分糟糕。中本聪在构建比特币用途时，他主要侧重于比特币在互联网上的使用。这当然是比特币的主要用途，但并非唯一用途。他从未想过，可以去咖啡店用比特币付钱。

了解了虚拟货币的历史之后，我们可能还存在一个疑问："为什么中本聪要匿名?"有许多可能的原因。首先，也许他就是喜欢这样。许多小说家都选择匿名，像班克西（Banksy）这样的涂鸦艺术家也一直不公开身份。其实，在中本聪活跃的网络朋克社区和密码学邮件组，大家都普遍采用匿名方式发表文章。

此外，中本聪选择匿名可能还有法律上的顾虑。自由储备（Liberty Reserve）和电子黄金（e-Gold）这两家美国公司都因为非法洗钱惹上了麻烦。2006 年，自由储备的创始人之一担心被指控洗钱，逃离美国。电子黄金的创始人一直待在美国，但其中一位创始人被指控洗钱并最后认罪。这一事件正好发生在中本聪创建比特币网站并公开讨论比特币的前夕。纵然如此，许多人都创立过虚拟货币系统，没有人因为法律顾虑而选择匿名。所以很难说这到底是不是他选择匿名的根本原因。

值得注意的是，我们之前提过，电子现金的一些技术是有专利保护的。网络朋克运动担心实施电子现金系统会侵犯这些专利。事实上，有人曾在网络朋克邮件组发表文章，建议由一群匿名的程序员来架设电子现金系统，这样，即使侵权，也查不出是谁。但是，比特币的设计与电子现金的专利差别很大，很难判定比特币侵犯了它的专利权，也许中本聪选择匿名只是比较谨慎。又或者，他是受到网络朋克社区里程序员匿名的启发。

许多人认为中本聪选择匿名是出于个人安全方面的考虑。众所周知，他早期时挖矿获得大量比特币，时至今日，比特币的巨大成功也就给他带来了巨额财富。我认为这个原因是可能的。毕竟，选择匿名不是一时的决定，而是一贯的风格。尽管如此，这可能还不是他一开始就选择匿名的原因。当他首次使用中本聪这个化名时，他还没有发布白皮书和源代码，很难想象他那时就能够预测到比特币后来会取得如此巨大的成功。其实，在早期，中本聪对比特币的未来持乐观且谨慎的态度。他明白许多之前的尝试都失败了，比特币最终也可能失败。

结语

与之前的失败尝试相比，比特币的成功令人瞩目。它有许多优秀的创新，例如区块链和去中心化实现用户之间直接交易的模式等。它能够有效地确保用户的匿名性，虽然还做得不够完美。我们将在第 6 章详细了解保密性。比特币的保密性从某种意义上来说做得不如数字现金那么好，但从另一个角度来看，它的保密性要更强。因为在数字现金系统，只有消费者能够匿名，商家则不能。比特币为消费者和商家（不管是消费者还是商家）提供了同等程度的保密性。

把比特币和我们之前讨论过的虚拟货币系统进行对比，我学习到的经验教训是：第一，遇到困难时不要轻易放弃。20 年来，人们在开发虚拟货币的道路上一直失败，但这并不意味着永远开发不出一套成功的体系。第二，要愿意折中妥协。如果你想把保密性和去中心化功能做到完美和极致，可能就必须牺牲其他的功能。回顾比特币的发展史，它找到了一个完美的平衡点。它的保密性不够完美，需要用户连接到点对点的网络，但用户愿意接受这样的设定。

最后，众志成城。比特币吸引了一批具有激情的用户和开发者，他们愿意为开源技术出一分力，这与之前由公司开发的虚拟货币很不一样，后者的支持者只是公司内部员工而已。比特币如今的成功很大一部分是因为它拥有一个庞大的生机勃勃的支持群体，他们共同推动科技的发展，招徕客户，说服商家采用它。

延伸阅读

一篇关于虚拟货币架构的综述，浅显易懂，侧重实践：

P. Wayner. *Digital Cash*: *Commerce on the Net* (2nded). Waltham, MA: Morgan Kaufmann, 1997.

从密码学角度看电子现金系统（第一章）和微支付（第七章）：

B. Rosenberg, ed. *Handbook of Financial Cryptography and Security*. Boca Ra-

ton, FL: CRC Press, 2011.

虽然不是乔姆最早一篇关于电子现金的论文，但这篇是公认的最富有创造性的论文。它的模式成为后来类似论文竞相模仿的对象：

D. Chaum, A. Fiat, and M. Naor. "Untraceable Electronic Cash." In *CRYPTO 88: Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*. London: Springer Verlag, 1990.

运用现代密码学技术来提升乔姆－菲亚特－纳欧尔体系效率的论文有许多篇，这是其中最重要的一篇：

J. Camenisch, S. Hohenberger, and A. Lysyanskaya. "Compact E-cash: Theory and Applications of Cryptographic Techniques," 2005.

对金融市场和金融构想，包括对 Mondex 电子钱包体系进行的一些比较实用的安全性分析：

R. Anderson. *Security Engineering*, second edition. Hoboken, NJ: Wiley, 2008.

乔姆的电子现金构想的实施纲要：

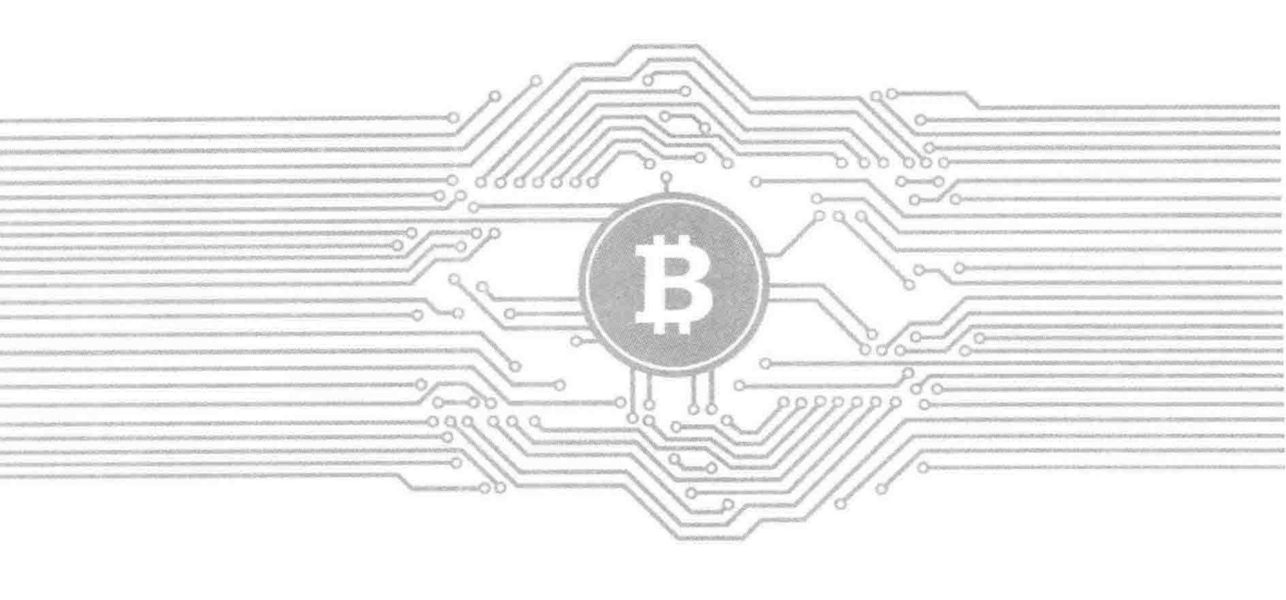
B. Schoenmakers. "Security Aspects of the Ecash Payment System." In *State of the Art in Applied Cryptography*. New York: Springer, 1997.

这两篇论文曾被中本聪在比特币白皮书中引用，被运用于比特币的设计中：

A. Back. "Hashcash—A Denial of Service Counter-Measure," 2002. Available at hashcash. org/papers/hashcash. pdf.

S. Haber and W. S. Stornetta. "Secure Names for Bitstrings." CCS, 1997.

CONTENTS | 目录



第1章

密码学及加密货币概述

所有货币都需要通过某种方式控制供给，并需要实施各种安全属性以防止欺骗行为发生。就法定货币而言，中央银行这样的机构控制货币供给，并在实体货币上加上防伪标识，这些安全属性提升了攻击货币的门槛和难度，但并非不可能伪造。最终，执法部门仍需要介入，以防止货币系统规则受到破坏。

加密数字货币也必须采取安全措施，以防御破坏系统状态的行为，同时加密数字货币还需要防止"混淆"，即对不同的人说出相互矛盾的话。例如，如果爱丽丝（Alice）让鲍勃（Bob）确信她向他支付了一个数字币，她就不能再说服卡罗尔（Carol），也给她支付同一个数字币。加密数字货币与法定货币的不同在于，其安全规则需要完全通过技术手段实现，而非依赖于中央机构。

顾名思义，加密货币着力采用密码技术。密码学提供一个将加密货币体系规则编码到系统本身的机制，我们不但可以利用密码学防止对系统的干扰，并且能够避免混淆，也能用其将新货币单位创造规则编码到数学协议中。为了能够深刻理解加密数字货币系统，我们需要首先探究该系统所依赖的密码学基础。

密码学是一个高深的学术领域，用到了很多不被大众所知的数学理论，并且其理论也比较复杂。幸运的是，比特币只运用到了密码学中少量相对较为浅显的一些理论。在本章中，我们会特别讨论一下密码学中的哈希算法（Hash）和数字签名（digital signature）技术，这两个基本概念对构建一个加密数字货币系统非常关键。在后面的章节中，我们会介绍一些更复杂的密码学理论，例如零知识验证（zero-knowledge proof），这个概念被应用到了对比特币网络的拓展和改进之中。

在学习了必要的密码学基础之后，我们将讨论如何用这些密码学基础构建一个加密数字货币系统。在本章末尾，我们会列举一些简单的加密货币案例，来阐明我们在设计中遇到的挑战。

1.1　密码学哈希函数

我们需要理解的第一个密码学的基础知识是**密码学哈希函数**，**哈希函数**是一个数学函数，具有以下三个特性：

- 其输入可为任意大小的字符串。
- 它产生固定大小的输出。为使本章讨论更具体，我们假设输出值大小为256位，但是，我们的讨论适用于任意规模的输出，只要其足够大。
- 它能进行有效计算，简单来说就是对于特定的输入字符串，在合理时间内，我们可以算出哈希函数的输出。更准确地说，对应n位的字符串，其哈希值计算的复杂度为O（n）。

这些特性定义了一般哈希函数，以这个函数为基础，我们可以创建数据结构，例如哈希表。我们将只专注于加密的哈希函数，要使哈希函数达到密码安全，我们要求其具有以下三个附加特性：（1）碰撞阻力（collision-resistance）；（2）隐秘性（hiding）；（3）谜题友好（puzzle-friendliness）。

我们会仔细研究这些特性，并会逐步阐释我们为什么需要这样的函数。学习过密码学的读者可能会注意到，我们这里对于哈希函数的论述与一般的密码学课程会有所不同，特别是关于谜题友好。在一般密码学中，谜题友好并非加密的哈希函数的一般要求，却对加密数字货币这一特性非常有用。

特性1：碰撞阻力

加密的哈希函数的第一个特性是它要具有碰撞阻力。这里的碰撞指对于两个不同的输入，产生相同的输出。如果对于哈希函数H（·），没有人能够找到

碰撞，我们则称该函数具有碰撞阻力（见图 1.1）。即：

> **碰撞阻力** 如果无法找到两个值，x 和 y，x≠y，而 H（x）=H（y），则称哈希函数 H 具有碰撞阻力。

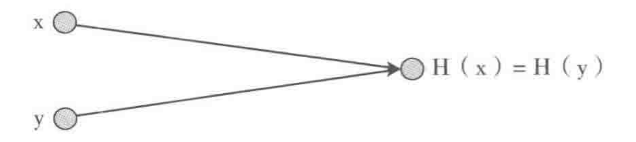


图 1.1 哈希碰撞

注：x 和 y 分别是不同输入，当作为哈希函数的输入时，会产生相同的输出。这时我们就说这个函数是哈希碰撞的。

注意，我们说没人能找到碰撞，并不表示不存在碰撞。事实上，通过简单的计数论证（counting argument），我们可以证明碰撞的确存在。哈希函数的输入空间包含所有长度的任意字符串，但输出空间则只包含特定固定长度的字符串。因为输入空间比输出空间大（输入空间是无限的，而输出空间是有限的），一定会有输入字符串映射到相同的输出字符串。实际上，根据鸽巢原理（Pigeonhole Principle），我们可以得出，必然会有大量可能的输入被映射到任意特定输出（见图 1.2）。

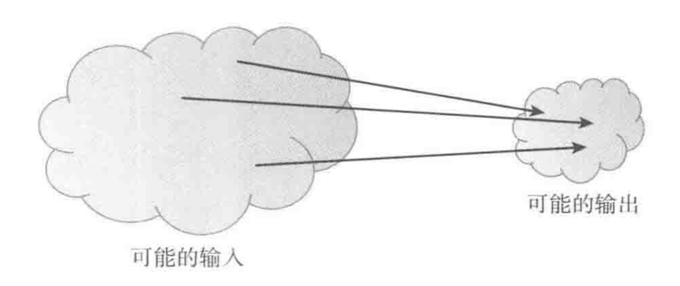


图 1.2 必然的碰撞

注：因为输入的数量超过输出的数量，我们可以确定某一个输出肯定对应多个输入。

而更糟糕的是，对于加密的哈希函数，我们虽然说应该找不到碰撞，但有些方法是能保证找到碰撞的。考虑以下对应于一个 256 位输出大小的哈希函数，选择 $2^{256}+1$ 个不同数值，计算每个数的哈希值，并检查是否有两个相等的输出。因为我

们这里选择的输入多于输出，因此在应用哈希函数时，一些数对必将产生碰撞。

使用上述方法一定能找到碰撞。但如果我们随机性地选择输入，并计算哈希值，我们在检验第 $2^{256}+1$ 个输入之前便很可能找到碰撞。实际上，如果我们随机选择 $2^{130}+1$ 个输入值，找到至少两个等同哈希值的概率为 99.8%。仅仅通过检验可能输出数量的平方根次数，便大体能找到碰撞，这一事实在概率学中被称为是生日悖论（birthday paradox）①。

这个碰撞检测算法对每个哈希函数都有效，但是它的问题是其计算需要花很长很长时间才能完成。对于一个 256 位输出的哈希函数来说，最坏的情况是你需要进行 $2^{256}+1$ 次哈希函数计算，平均次数为 2^{128} 次，这简直是一个天文数字——如果一台电脑每秒计算 10 000 个哈希值，计算 2^{128} 个哈希值，需要花 10^{27} 多年时间！换个角度，我们可以说，如果人类制造的每台电脑在整个宇宙起源时便开始计算，到目前为止，它们找到碰撞的概率仍然无穷小，比下两秒钟地球将被大陨石摧毁的概率还要小得多。

因此，为了寻找一个任意的哈希函数的碰撞，我们只是有了一个一般，但并不实用的算法。一个更艰涩的问题是：有没有其他的方法，可以用于对于某一特定哈希函数找到碰撞？也就是说，虽然一般的碰撞测试算法不适用，但仍可能有其他的算法，可以有效地找到某个哈希函数的碰撞。

以下面的哈希函数为例：

$$H(\chi) = \chi \bmod 2^{256}$$

这个函数接受任何长度的输入，产生一个固定大小输出（256 位），且能进行有效计算，因此符合我们对哈希函数的要求。但是对于这个函数，我们确实具备一个有效的能够寻找碰撞的方法。注意，这个函数仅返回输入的最后 256 位，因此，数值 3 和 $3+2^{256}$ 就构成了碰撞②。这个简单的例子表明，虽然我们的一般碰撞

① 生日悖论是指，如果一个房间里有 23 个或 23 个以上的人，那么至少有两个人的生日相同的概率要大于 50%。这就意味着在一个典型的标准小学班级（30 人）中，存在两人生日相同的可能性更高。对于 60 或者更多的人，这种概率要大于 99%。——译者注

② 3 和 $3+2^{256}$ 对 2^{256} 求余数之后，结果都是 3。——译者注

测试方法在实践中不可行，但至少对于某些哈希函数，存在有效的测试碰撞的方法。

但对于某些哈希函数，我们无法确认识别碰撞的有效方法是否存在，我们只是怀疑这些函数具有防碰撞特性，但是我们已经证明，世界上没有哈希函数具有防碰撞特性。我们实践中依赖的加密的哈希函数仅仅是人们经过不懈努力之后暂未成功找到碰撞的函数。因此，我们选择相信那些加密的哈希函数具有哈希阻力（在某些情况下，如之前的 MD5 哈希函数，在多年的努力之后最终找到了碰撞，导致该函数在实践中被逐渐淘汰，最终被弃用）。

应用：信息摘要

现在我们知道什么是碰撞阻力了，我们自然会问：碰撞阻力有什么用途？以下就是一个应用：哈希函数 **H 具有碰撞阻力**，x 和 y 是两个不同的输入，那么可以假设它们的哈希函数 H（x）和 H（y）也不同——如果已知 x 和 y 不同，但哈希值相同，那么 H 具有碰撞阻力的假设就不成立了。

这个论证使我们可以将哈希输出作为**信息摘要**（message digest）。以 SecureBox 为例，SecureBox 是一个允许用户上传文件，并保证文件被完整下载的线上文件存储系统。假设爱丽丝上传了很大的文件，并希望能够在之后下载时确认她下载的文件与她上传的文件相同。一种方法是将整个文件进行本地存储，并直接将其与下载文件对比。如果这样可行，那么将文件上传便显得毫无意义，倘若爱丽丝需要使用本地文件副本以保证其完整性，她可以直接使用本地副本。

无碰撞哈希函数为这个问题提供了简单有效的解决方法，爱丽丝只需要记住原文件的哈希值，从 SecureBox 下载文件后，她可以计算下载文件的哈希值，并与原文件哈希值进行对比。如果哈希值相同，那么爱丽丝可以说该文件就是她上传的那一个，但是如果不同，她则可以确定文件被破坏了。记录哈希值可以帮助爱丽丝检测文件在传输过程中，或在 SecureBox 服务器上是否产生了意外损坏，或者检测文件是否受到服务器的蓄意修改。保证主体不受其他实体的恶意行为侵害，这正是密码学的核心。

这里的哈希函数对于一个信息生成固定长度的摘要，或生成了简明总结，这为我们提供了一种记住之前所见事物，并在今后认出这些事物的有效方法。

虽然整个文件可能非常大，存储规模达数 G，但其哈希值的长度固定。例如，哈希函数为 256 位。这样做，极大地降低了存储要求。在本章后面及整本书中，我们都会看到哈希作为信息摘要的应用。

特性 2：隐秘性

我们希望哈希函数拥有的第二个特性是其**隐秘性**。隐秘性保证，如果我们仅仅知道哈希函数的输出 y = H（x），我们没有可行的办法算出输入值 x。问题是，上述的表示形式不一定是正确的。考虑以下简单的例子：我们做一个抛硬币的实验，如果抛硬币结果为正面，我们会宣布字符串哈希为“正面”；如果结果为反面，我们会宣布字符串哈希为“反面”。

然后我们问我们的对手，在他没有见到抛硬币，而只见到哈希函数的输出的前提下说出哈希函数的输入字符串（很快我们就知道为什么要玩这个游戏了）。为了回答问题，对手会简单计算“正面”字符串的哈希值及“反面”字符串的哈希值，然后对手便可以知道他得到的是哪一个。这样，只需要几步，对手就能反解出输入值。

对手能够猜出字符串，这是因为 x 只有两个可能，他可以很轻易地将两个可能对应的哈希值计算出来。为了能够实现隐秘性，我们需要 x 的取值来自一个非常广泛的集合，也就是说，仅仅通过尝试几个特定的 x，就能找到输出值的方式将不会发生了。

现在的问题是：在类似抛硬币的“正面”、“反面”实验中，如果我们想要的反解的输入值并非来自分散的集合，我们是否还能得到隐秘性？幸运的是，对于这个问题答案是肯定的！我们甚至能够通过与另一个较为分散的输入进行结合，而将一个并不分散的输入进行隐秘。现在我们可以更精确地表示隐秘的含义了（双竖线 ‖ 为连接符号，代表把一系列事件、事情等联系起来）。

> **隐秘性**　哈希函数 H 具有隐秘性，如果：当其输入 r 选自一个高阶最小熵（high min-entroy）的概率分布，在给定 H（r ‖ x）条件下来确定 x 是不可行的。

在信息论中，**最小熵**是用于测试结果可预测性的手段，而高阶最小熵这个概念比较直观描述了分布（如随机变量）的分散程度。具体来说，在从这样分

布中取样时，我们将无法判定取样的倾向。举个具体的例子，如果r是从长度为256位的字符串中随意选出的，那么任意特定字符串被选中的概率为$1/2^{256}$，这是一个小到几乎可以忽略的取值。

应用：承诺

现在来看一下隐秘性的应用。具体来说，我们把想做的事情称为**承诺**（commitment）。这里承诺是一个数字化过程，可以类比为以下动作：首先选定一个数字，将数字装进信封，然后将该信封放到一个人人都看得到的桌子上。这样做以后，可以说你就信封里的数字做出了承诺，在打开信封前，虽然你已经做出了承诺，对其他人来说它还是秘密。在之后，你可以打开信封，来展示承诺的数值。

承诺协议　一个承诺协议方案由两个算法构成：

- com：=commit（msg，nonce），承诺函数将信息（msg）和一个临时随机数（nonce）作为输入，输出就是一个“承诺”。
- verify（com，msg，nonce），验证函数将某个承诺输出（com）、临时随机数（nonce）及信息（msg）作为输入，如果com = = commit（msg，nonce），则返回“真”（true）；反之则返回“假”（false）。

我们要求以下两个安全特性要成立：

- 隐秘性：已知com，没有可行的方法找到msg。
- 约束性：没有可行的办法找到两组（msg，nonce）和（msg’，nonce’），msg≠msg’，而commit（msg，nonce） = =commit（msg’，nonce’）。

为了使用承诺方案，我们首先需要产生一个临时随机数。然后将这个临时随机数与承诺信息msg一起代入承诺函数，计算承诺函数输出值com，然后公布该输出。这个过程就如同将封好的信封放到一个人人能看到的桌上那样。之后，如果我们希望展示之前的承诺值，我们首先公布用于产生承诺的临时随机数，并公布信息msg。此时任何人都可以验证这时公布的msg是否为之前承诺，这个

阶段就如同打开信封。

> 对于每次的承诺值，你都需要选择新的随机值，这一点很重要。在密码学中，术语 nonce 是指，该取值只能使用一次。

以上两个安全特性决定了这一算法就如同密封及打开信封。第一，如果仅仅知道 com，即承诺函数的输出，就如同只看信封并不能得到信息内容；第二就是约束性，这就保证了你一旦承诺信封内的内容，就不能再改变主意。也就是说，我们无法找到两个不同的信息，当你在承诺一个信息后，而又声称你承诺了另一个信息。

我们如何在承诺协议中保证隐秘性和约束性这两个性质成立呢？在讨论这一点之前，我们需要讨论如何执行承诺方案。我们可以通过使用加密的哈希函数来达到目的，考虑如下承诺协议实施方案：

commit（msg，nonce）：= H（nonce ‖ msg）

其中，nonce 为长度为 256 位的临时随机数。

为承诺一段消息，我们首先生成一个 256 位的临时随机数，然后将这个临时随机数与信息链接，并返回这个链接值的哈希值，来作为承诺输出。为了便于验证，我们还要设定其他人来计算一下临时随机数与信息链接之后的哈希值，比对一下计算结果是否与承诺输出相同。

再来看一下我们的承诺方案要求的两个特性，如果我们将承诺和验证换成 H（nonce ‖ msg），那么这些特性就变成：

- 隐秘性：已知 H（nonce ‖ msg），没有可行方法找到 msg。
- 约束性：没有可行方法找到两对（msg，nonce）和（msg’，nonce’），msg≠msg’，而 H（nonce ‖ msg）==H（nonce’ ‖ msg’）。

承诺的隐秘特性正是我们要求哈希函数要具备的隐秘性，如果将一个解密钥匙选定为 256 位的随机值，那么由隐秘性得出，如果解密钥匙与信息链接，那么仅仅从哈希函数的输出中恢复信息就是不可行的。约束性隐含在哈希函数

的碰撞阻力特性中[1]，如果哈希函数具有碰撞阻力，那么我们将不能找到不同的 msg 及 msg' 值，而 H（nonce ‖ msg） = H（nonce' ‖ msg'），如果这种情况发生，将构成碰撞。

因此，如果哈希函数 H 具有碰撞阻力及隐秘性，从安全特性上来讲，这个承诺方案将有效。

特性3：谜题友好

哈希函数需要的第三个安全特性为谜题友好特性。这一特性较为复杂，我们首先解释该特性的技术要求，然后通过举例来阐释该特性的意义。

直觉上，谜题友好可以这样解释，如果有一个人想找到 y 值所对应的输入，假定在输入集合中，有一部分是非常随机的，那么他将非常难以求得 y 值对应的输入。

> **谜题友好**　如果对于任意 n 位输出值 y，假定 k 选自高阶最小熵分布，如果无法找到一个可行的方法，在比 2^n 小很多时间内找到 x，保证 H（k ‖ x） = y 成立，那么我们称哈希函数 H 为谜题友好。

应用：搜索谜题

现在，让我们试想一个应用以阐释谜题友好特性的意义。在这个应用中，我们将建立一个搜索谜题，该谜题是一个需要对庞大空间进行搜索，才能找到解决办法的数学问题。该搜索谜题没有捷径，也就是说除了搜索庞大的空间来进行求解，别无他法。

> **搜索谜题**　搜索谜题构成：
>
> - 一个哈希函数 H。
> - 从高阶最小熵分布选出的一个取值，id（我们称其为谜题 ID）。

---

① 结论反之不成立，就是说，我们可以找到碰撞，但都不是满足 H（nonce ‖ msg） = = H（nonce' ‖ msg'）意义下的碰撞。例如，你可以对于同一个信息来产生满足同一承诺的随机数，但这里的哈希函数不具备碰撞阻力特性。

- 目标集合 Y。

该谜题的解决方法为一个解，x，应该满足以下公式：

$$H\ (id \parallel x)\ \in Y$$

这个直觉是：如果 H 有一个 n 位输出，那么它的可能取值有 2^n 个。解决这个谜题要求找到一个位于集合 Y（通常比所有输出值集合小很多）内的输出值，Y 的大小决定了谜题的难度。如果 Y 是所有 n 位字符串的集合，这个谜题就毫无意义。然而，如果 Y 只有一个元素，那么这个谜题难度最大，谜题 ID 取自高阶最小熵分布，这个事实保证了求解无捷径。反过来，如果该 ID 的确定性很高，那么有人可能会作弊，比如通过使用该 ID，事先对谜题进行求解。

如果一个哈希函数具备谜题友好特性，这就意味着对于这个谜题没有一个解决策略，比只是随机地尝试 x 取值会更好。因此，如果我们要把谜题做成很难解决是可以的，只要我们能用适合的随机方式生成谜题 ID。当我们讨论比特币采矿（是一种搜索谜题）时会采用这一思路。

安全哈希算法

我们讨论了哈希函数的三个特性及其相应的应用。现在，让我们讨论本书中将会大量用到的一个哈希函数，**安全哈希算法**（Secure Hash Algorithm 256，简称 SHA－256）。哈希函数有很多，但 SHA－256 是一个主要被比特币世界采用，并且效果还很不错的哈希函数。

回想一下，我们要求哈希函数可以用于任意长度输入。幸运的是，只要我们能建立一个用于固定长度输入的哈希函数，然后通过一般方法，就可以将接受固定长度的哈希函数转化为可接受任意长度输入的哈希函数，我们称这个转换过程为 **MD**（Merkle-Damgard）**变换**，SHA－256 是采用这种变换方法的常用哈希函数之一。在通用术语中，这种基础型，可用于固定长度，具备碰撞阻力的哈希函数被称为是**压缩函数**（compression function）。经过验证，如果基本压缩函数具有碰撞阻力的特性，那么经过转换而生成的哈希函数也具有碰撞阻力。

MD 变换很简单。比如压缩函数代入长度为 m 的输入值，并产生长度短一

些为 n 的输出值。哈希函数的输入（可为任意大小）被分为长度为 m－n 的区块。MD 变换运作过程如下：将每个区块与之前区块的输出一起代入压缩函数，注意，输入长度则变为（m－n）+n=m，也刚好就是压缩函数的输入长度。对于第一个区块而言，之前没有的区块，我们需要选取一个初始向量（见图 1.3）。每次调用哈希函数，这个数字都会被再一次使用，而在实践中，你可以直接在标准文档中找到它。最后一个区块的输出也就是你返回的结果。

SHA－256 函数利用了这样的一个压缩函数，这个压缩函数把一个 768 位的输入压缩成一个 256 位的输出，每一个区块的大小是 512 位。我们可以通过图 1.3来理解 SHA－256 的工作过程。

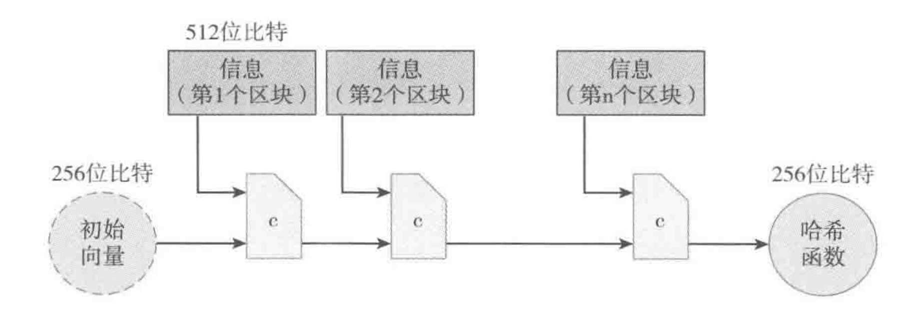


图 1.3　SHA－256 哈希函数简化图

注：SHA－256 利用 MD 变换把一个固定输入的防止碰撞的压缩函数变换成一个接受任意长度输入的哈希函数。通过初始化向量的补位，可以把输入变成 512 位比特的整数倍。

截至目前，我们已经讨论了哈希函数、密码学上使用具备特性的哈希函数、这些特性的应用，以及在比特币世界中使用的一类特殊的哈希函数。在下面的章节中，我们将讨论通过哈希函数来构建比特币网络中的更为复杂的数据结构。

哈希函数建模

哈希函数是密码学中的瑞士军刀：它们在众多各具特色的应用中找到了一席之地。这种多功能性的另一面是，为了保证安全，不同的应用会要求不同的哈希函数特性。事实已经证明，要确定一系列哈希函数特性以全面达成

可证安全极度困难。

本书中，我们会选出在比特币和其他加密数字货币中，对哈希函数使用方式很重要的三个特性。即使在这个范围内，并非所有这些特性对哈希函数的每一次使用都有必要。比如，我们之后会看到，谜题友好只在比特币采矿中具有重要性。

安全系统设计师常常会放弃，并且把哈希函数建立成对于任意一个可能的输入，都会得到一个独立的随机输出的函数。这种使用“随机预言模式”来证明安全的做法在密码学中仍具争议。不论在这个辩论中你的立场如何，在建立安全系统时，当我们应用哈希函数基本特性，推论如何减少安全特性的数量，都是宝贵的智力训练。本章的目的便是帮你学习这一项技能。

1.2 哈希指针及数据结构

本节将讨论哈希指针（hash pointer）及其应用。哈希指针是一种数据结构，这种数据结构在我们即将讨论的很多系统中都很有用。简单来说，哈希指针是一个指向数据存储位置及其位置数据的哈希值的指针。一个普通的指针可以告诉你数据存储的位置，哈希指针不但可以告诉你数据存储的位置，并且还可以给你一种方式，让你验证数据没有被篡改过（见图 1.4）。

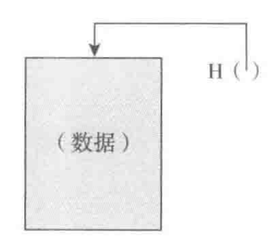


图 1.4　哈希指针

注：哈希指针是一个不但可以指向数据存储的位置，还可以明晰某个时间戳下该数据的哈希值的指针。

我们可以利用哈希指针构建各种各样的数据结构。为求直观，我们可以把原来用普通指针实现的数据链表和二叉查找树通过哈希指针来实现。

区块链

如图 1.5 所示，我们通过哈希指针构建一个链表，我们将这个数据结构称为**区块链**（block chain）。在普通链表中有一系列区块，每个 a 区块既有数据也有一个指向上一个区块的指针。而在区块链中，上一个区块指针被置换为哈希指针。因此，每个区块不仅能告诉我们上一个区块的值在哪里，还包含了该值的摘要，使我们能够验证那个值没有改变。我们存储链表头部（the head of list），即一个普通的哈希指针指向最近使用的数据区块。

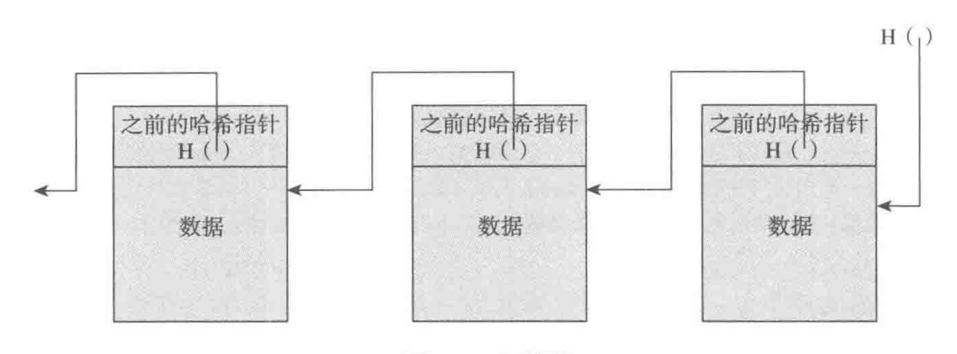


图 1.5　区块链

注：通过哈希指针而不是普通指针构建的一个链表，我们把这个链表称为区块链。

区块链的一个应用就是“防篡改日志”。也就是说，我们要建立一个存储很多数据的日志数据结构，使我们能将数据附加到日志的末尾。但是如果有人篡改日志前面的数据，我们可以检测到。

要理解区块链如何实现这一防篡改特性，我们先看一下如果对手要篡改区块链中间的数据会发生什么。具体来说，通过这种方式，对手的目的是让只记得区块链头部哈希指针的人无法检测到篡改行为。为达到这个目标，对手会改变某区块 k 的数据。既然数据已经被改变，区块 k + 1 的哈希值（即整个区块 k 的哈希值）将不会匹配。记住，因为哈希函数具有碰撞阻力，我们可以确定新的哈希值与改变后的内容不会匹配。因此，我们会检测到区块 k 中的新数据以

及区块 k +1 中的哈希指针的不一致性。当然，对手可以继续尝试，并通过篡改下一个区块的哈希值掩盖这次篡改。他可以一直这样做，但是当他到达链表的头部时，这个策略将会失败。具体来说，只要我们将链表头部的哈希指针存储在对手无法改动的地方，对手将不能做到在不被检测到的前提下，篡改任何区块（见图 1.6）。

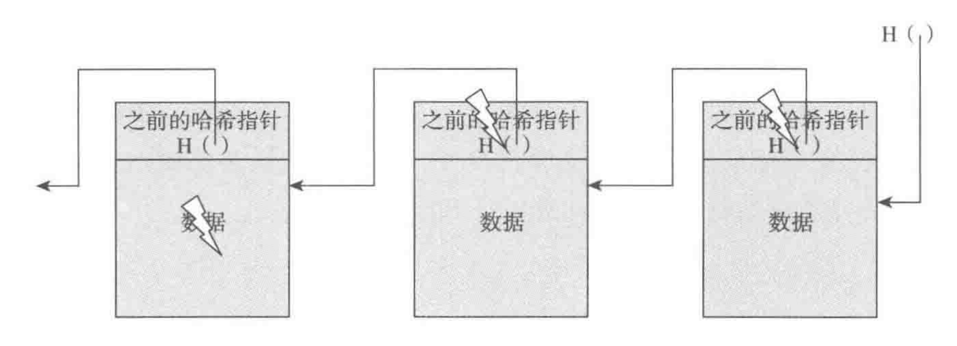


图 1.6　防篡改日志

注：如果对手修改了区块链中的任意部位的数据，那么将会导致下一个数据块的哈希指针不正确。如果我们锁定区块链的头部数据，那么即使对手修改了所有哈希指针使其与修改过的数据一致，那么他也无法修改头部数据，从而我们就可以检测到篡改行为。

这样做的结果是，如果对手想要篡改区块链中任意地方的数据，为了保证整个内容一致，他需要篡改所有的哈希指针直至最开始的地方。他最终将碰到障碍，因为他不能篡改链表头部的指针。这样，我们便知道，仅通过记住一个哈希指针，我们就基本记住了整个链表的防篡改哈希值。因此，我们可以搭建一个包含很多区块的区块链网络，链表头部的哈希指针被称作**创世区块**（genesis block）。

你可能已经注意到了，区块链的结构与我们上一节见到的 MD 变换类似。的确，它们很相似，同一个安全论证对于两者都适用。

梅克尔树

另一个我们可以用哈希指针建立的有用的数据结构是二叉树。使用哈希指针的二叉树也叫作**梅克尔树**（Merkle trees），以其发明者拉尔夫 · 梅克尔

（Ralph Merkle）的名字命名。如图 1.7 所示，假设我们有很多包含数据的区块，这些区块就构成了树的叶子（节点）。我们将这些数据区块两两分组，然后为每一组建立一个有两个哈希指针的数据结构，每个指针对应一个区块，这些数据结构就构成了树的下一个层次。我们轮流将这些区块组两两分组，为每一组建立一个包含每个区块组哈希指针的新的数据结构。以此类推，直到我们得到一个单一区块，即树根节点。

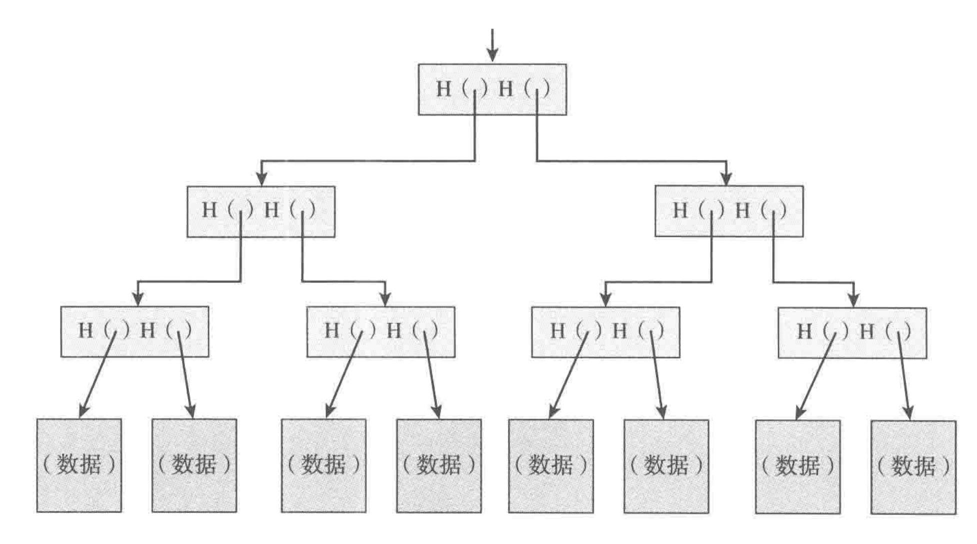


图 1.7　梅克尔树

注：在梅克尔树的数据结构中，所有的数据区块都被两两分组，指向这些数据区块的指针被存储在上一层的父节点（parent node）中，而这些父节点再次被两两分组，并且指向父节点的指针被存储在上一层的父节点中，一直持续这个过程，直到最后我们到达树的根节点。

如上所述，我们要记住树最前面的哈希指针。我们现在可以通过哈希指针回溯到列表中的任何位置，这让我们能保证数据确实未经篡改，正如我们在区块链见过的一样，如果对手篡改了树底部的一些数据区块，会导致上一层的哈希指针不匹配，即使他继续篡改这个区块，改动数据行为将最终传递到树的顶端，而此时，他将不能篡改我们存储的哈希指针。因此，同样地仅仅通过记住顶部的哈希指针，任何企图篡改任何数据的行为都会被检测到。

隶属证明

与我们之前建立的区块链不同，梅克尔树的另一个特点是它可以实现简洁的隶属证明。假设某人想要证明某个数据区块隶属于梅克尔树。同样地，我们只记住树根节点，然后他需要展示给我们数据块信息，以及从该数据区块通向树根节点的那些区块，我们可以忽略树的其余部分，这样做是因为这些区块已经足够让我们验证通往树根节点过程中所有的哈希值。其工作原理图解参见图 1.8。

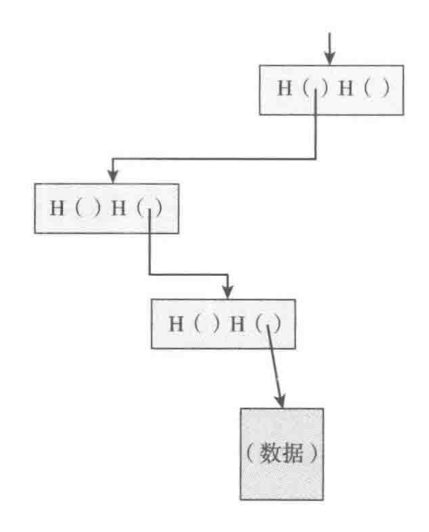


图 1.8　隶属证明

注：为了证明某个数据区块来自一个梅克尔树，我们只需要找到该数据区块到树根节点的路径。

如果整棵树上有 n 个节点，只需要展示约 log（n）个项目，因为每个步骤仅需要计算子区块的哈希值，验证过程需要时间约为 log（n）。因此，即使梅克尔树包含大量的区块，我们仍可以在相对较短时间内证明隶属关系。因此，验证需要花的时间和涉及空间（树节点）与 log（n）同级。

一个排序梅克尔树是把底层的数据通过某些排序得到的梅克尔树，这里排序规则可以是字母表排序、词典排序、数字化排序，或者其他约定的排序方式。

非隶属证明

有了排序梅克尔树，我们可以在一个对数复杂度的条件下验证某一个数据区块并非来自某梅克尔树。也就是说，我们可以证明某个特定区块不属于梅克尔树，而我们只是简单通过展示被验证区块之前的区块路径，以及被验证区块之后的区块路径，就可以达到目的。如果之前、之后两个区块在树上是连续的，那么这说明了被验证区块与该梅克尔树之间是非隶属关系。因为被验证区块确实隶属于梅克尔树，它需要在两个条目之间，而如果两个条目是连续的话，二者之间则并没有空间。

我们讨论过在链表及二叉树中使用哈希指针，但更广泛地说，我们可以在任何以指针为基础的数据结构中使用哈希指针，条件是数据结构不存在循环。如果数据结构中存在循环，那么我们将不能使所有哈希值得到匹配。想一下，在一个非循环的数据结构中，我们可以在靠近节点的地方开始，或者在没有指针的数据区块开始，计算其哈希值，然后从后往前进行计算。但是在一个有循环结构的网络中，并没有一个根节点，可以让我们去追溯。

因此，试想另一个例子，我们可以建立一个哈希指针定向的非循环图。

我们能够在该图中非常有效地验证隶属关系，同时也方便计算。这样的哈希指针使用方式是一个常见技巧，在分布数据结构中、在本章后面会讨论到的算法中以及本书中都会反复提到。

1.3 数字签名

在本节，我们将讨论**数字签名**（digital signatures）。数字签名是密码学中的第二个重要部分，该理论和哈希函数一起，为我们后面讨论加密货币奠定基础。数字签名被认为是对纸上手写签名的数字模拟。我们对数字签名有两个特性要求，使其与我们对手写签名的预期一致。第一，只有你可以制作你自己的签名，但任何看到它的人都可以验证其有效性；第二，我们希望签名只与某一特定文

件发生联系，因此该签名不能用于表明你同意或支持另一份不同的文件。对于手写签名来说，第二条就如同确保别人不能将你的签名从一份文件上剪下来，贴到另一份文件的末尾那样。

那我们如何通过密码学来构建这些性质呢？首先，让我们把之前的直观讨论说得更具体一些，以便今后可以更好地论证数字签名方案，并讨论其安全特性。

数字签名方案

数字签名方案由以下三个算法构成：

- （sk，pk）：=generateKeys（keysize） generateKeys 方法把 keysize 作为输入，来产生一对公钥和私钥。私钥 sk 被安全保存，并用来签名一段消息；公钥 pk 是人人都可以找到的，拿到它，就可以用来验证你的签名。
- sig：=sign（sk，message） 签名过程是把一段消息和私钥作为一个输入，对于消息输出是签名。
- isValid：=verify（pk，message，sig） 验证过程是通过把一段消息和签名消息与公钥作为输入，如果返回的结果是真，证明签名属实；如果返回的结果为假，证明签名消息为假。

我们要求以下两个性质有效：

- 有效签名可以通过验证，即：

  verify（pk，message，sign（sk，message））==true
- 签名不可伪造。

我们注意到 generateKeys 和 sign 都可以采用随机算法。的确，generateKeys 最好是随机的，因为它需要为不同的人生成不同的密钥，而 verify 则需要是确定的。

现在，让我们更详细地检验我们要求数字签名方案具备的两个特性。第一个特性很直接，那就是有效的签名必须通过验证。如果我用我的密钥 sk 签署了一条消息，之后有人试图通过使用我的公钥 pk 验证关于同一条消息的签名，该签名必须证实为正确。这个特性是对签名有效的最基本要求。

不可伪造性。第二个要求计算上不可能伪造签名。也就是说，知道你公钥并看到你在某些信息上签名的对手，不能伪造他还未见过的你在其他信息上的签名。这一不可伪造特性类似于我们与对手之间在进行一场游戏，游戏的使用在密码安全证明中很常见。

在不可伪造性游戏中，对手会声称他可以伪造签名，而挑战者会测试他所说的话（见图 1.9）。我们做的第一件事是使用 generateKeys 方法生成一个密钥，以及相应的公共验证公钥，我们将密钥交给挑战者，然后将公钥交给挑战者以及对手。因此，对手只知道公共信息，而他的任务是试图伪造一条信息。挑战者知道密钥，因此他可以签名。

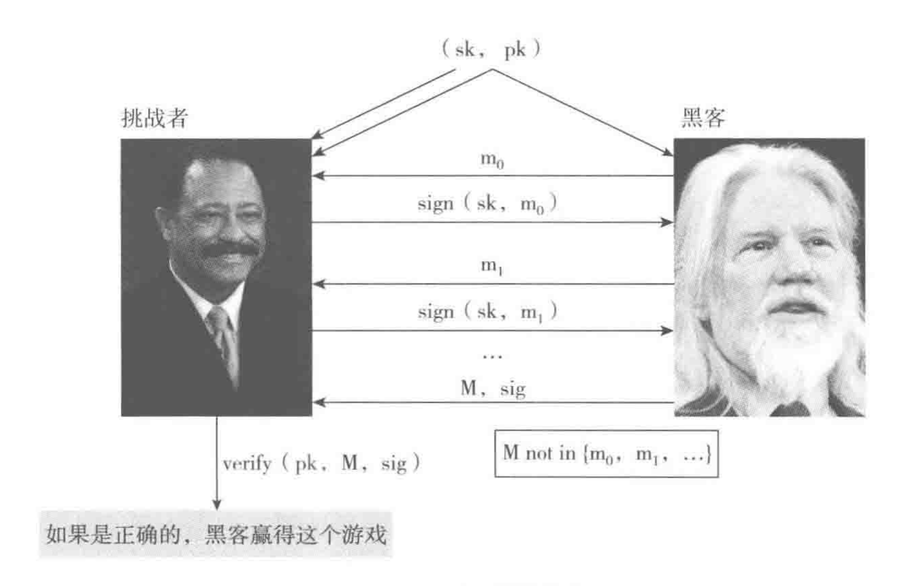


图 1.9　不可伪造性游戏

注：不可伪造性游戏是对手（黑客）和挑战者一起玩这样一个游戏：如果黑客可以在一个之前没有见过的消息上进行签名，那么黑客就赢得这个游戏；反之，如果黑客做不到，挑战者就赢得游戏，从而可以证明这个数字签名方案是不可伪造的。

直观来看，这个游戏的设定与真实世界条件一致，现实中的攻击者很可能可以从潜在受害者的很多不同文件中看到有效签名，攻击者甚至还可能操控受害者签署一份看起来无害但对黑客有利的文件。

为了将这一点建模到我们的游戏中，我们将允许黑客选择一些文件的签名，

不限时长，只要猜测的数量合情。合情猜测数量的意思是，我们允许攻击者尝试猜测的次数高达百万，但数量高达 2^{80} 就不行了。从渐进性角度来说，我们允许攻击者多次尝试，尝试次数可以是一个密钥大小的多项式函数，但次数不能更多（例如攻击者不能以指数方式猜测）。

一旦攻击者满意他所看到的签名数量，那他就可以挑选某条信息 M，尝试在上面伪造签名。对 M 的唯一限制就是，它必须为攻击者之前未在之上看过签名的信息（因为很明显，攻击者可以发出他收到过的签名）。挑战者运行验证算法，以此确定攻击者生成的关于 M 信息签名在经过公共验证密钥验证后，是否属实。如果验证成功，攻击者赢得游戏。

不论对手使用什么算法，我们说签名方案不可伪造，当且仅当他成功伪造信息的机会非常小——小到我们可以假设在实践中从不会发生。

实践中的考量

要将算法概念转化为现实中可执行的数字签名机制，我们还需要考虑许多实际问题。例如，很多签名算法是随机的（特别是比特币使用的算法），因此我们需要随机性的良好来源。我们不能低估这一点的重要性，因为不良随机性会使你认为安全的算法变得不安全。

另一个实际问题是关于信息大小。在实践中，你能够签署的信息大小是有限制的，因为真实的方案将在位数长度有所限制的字符串中运行。有一个简单的方法可以解决这个限制：对信息的哈希值进行签署，而非对信息本身进行签署。如果我们使用输出值为 256 位的加密的哈希函数，那么我们可以有效地签署任何长度的信息，只要我们的签名方案能够签署 256 位的信息。如上所述，我们可以将信息的哈希值作为信息摘要，哈希函数具有碰撞阻力，因此这种方式是安全的。

我们后面会用到的另一个技巧是，可以对于哈希指针进行签署。如果你签署了哈希指针，那么该签名覆盖（或者说保护）整个结构——这不仅仅是哈希指针本身，还包括哈希指针指向的整个区块链。比如，如果签署了区块链末尾的哈希指针，其结果就是你有效地数字签署了整条区块链。

椭圆曲线数字签名算法

现在让我们来看一下具体的细节。比特币使用的数字签名方案叫作椭圆曲线数字签名算法（ECDSA）。ECDSA 为美国政府的标准，是早前 DSA① 算法利用了椭圆曲线的升级版。这些算法经过了数年的细致密码分析，且被普遍认为是安全的。

更具体地说，比特币使用 ECDSA 算法，而不是标准椭圆曲线“secp256k1”［预计提供 128 位安全保障，即打破这个算法的难度与执行 2^{128} 对称性密钥运算（如破解哈希函数）一样困难］。虽然这个曲线是公开标准，但除比特币以外鲜有使用，其他使用 ECDSA 的应用（如安全网络浏览时的 TLS② 密钥交换）通常都使用更常见的“secp256k1”曲线。这就是比特币的一个古怪之处，因为在比特币系统早期实施中被中本聪选定（参见原版前言），现在已很难改变。

我们不会详细地讨论 ECDSA 的原理，因为这涉及一些过于复杂的数学知识，且对于本书的其他内容没有太多帮助。如果你对 ECDSA 感兴趣，请参见本章末尾延伸阅读部分。虽然我们这么说，但对于了解各种参数也许会很有必要：

个人密钥：256 位
公钥（未压缩）：512 位
公钥（压缩）：257 位
待签名信息：256 位
签名：512 位

注意，严格来讲，虽然 ECDSA 只能签署 256 位的信息，但这存在问题，因为信息在签署之前总是已经经过哈希压缩，因此，任何大小的信息都能被有效签署。

使用 ECDSA 时，确保随机性良好来源至关重要，因为不良来源将可能导致

---

① DSA（Digital Signature Algorithm），电子签名算法。——译者注
② TLS（Transport Layer Security），传输层安全协议，用于在两个通信应用程序之间提供保密性和数据完整性。——译者注

密钥信息的泄露。这一点不难理解，如果你使用了不良随机来生成密钥，那么该密钥就可能不安全。但是 ECDSA 的古怪就在于，即使你仅仅只是在生成签名时使用了不良随机，而你使用的密钥完美无缺，你的个人密钥还是有可能泄露（熟悉 DSA 的人都知道这是 DSA 的古怪之处，但并不针对椭圆曲线）。接着游戏就结束了，如果你的个人密钥泄露，对手就可以伪造你的签名。因此，我们在实践中要特别注意使用良好随机来源，使用不良随机来源是安全系统的一个常见缺陷。

数字签名作为密码学基础，我们对其讨论就此结束。在下一节，我们将讨论对打造加密货币会带来帮助的一些数字签名应用。

加密货币及加密术

如果你一直在期待比特币使用的加密算法，我们可能会让你失望了，比特币并没有使用任何加密术，因为并没有加密的需要。加密术只是因为现代密码学而变得可能成为众多技术中的一个，很多技术（如承诺方案）在某种程度上隐藏信息，但是与加密术有所不同。

1.4 公钥即身份

让我们来看一下与数字签名并行的一个有用技巧，基本想法是从数字签名模式中拿出一个公共验证密钥，并将其与一个人或一个系统参与者的身份对等。如果你见到一条消息的签名被公钥 pk 正确验证，那么你可以认为 pk 就是在表达这条消息。你真的可以将公钥认为是参与者或者系统的一方，他可以通过签署声明而发布声明。从这个角度来说，公钥就是身份，让某人能为 pk 身份发声，他必须知道相应的密钥 sk。

将公钥视为身份的一个结果是，你可以随时制定新的身份——你可以简单

通过数字签名方案中的 generateKeys 程序，生成新的密钥对 sk 和 pk。pk 是你可以使用的新的公共身份，sk 是相应的密钥，只有你自己知道并可以让你代表身份为 pk 发声。在实践中，你可能会使用 pk 的哈希作为你的身份，这是因为公钥很大。如果是这样的话，为了验证消息来自你的身份，人们会需要验证：（1）你的身份确实是 pk 的哈希；（2）信息能经过公钥 pk 验证。

此外，在默认情况下，你的公钥 pk 基本上看起来是随机的，也并没有人能够通过检查 pk 发现你的现实身份（当然，一旦你开始使用这个身份发表声明，这些声明可能泄露信息，而让别人将你的真实身份与 pk 联系起来。我们很快会更详细地讨论这个问题）。你可以生成一个看起来随机的新身份，看起来像人群中的一张脸，但这些都只有你能够控制。

去中心化身份管理

公钥和私钥的体系，帮助我们引入去中心化的身份管理的理念。你可以自己作为用户注册，而无须到一个中央机构注册为系统用户。你不需要别人给你一个用户名，你也不需要告诉任何人你会使用什么名字。如果你想要新的身份，可以随时生成一个，而且想要多少就生成多少。如果你希望拥有五个不同的名字，没有问题！那就生成五个身份。如果你想匿名一阵子，你可以生成一个新的身份，使用一段时间，然后弃之不用。有了去中心化身份管理，所有这一切都变得可能。事实上，这就是比特币对待身份的方式。这些身份在比特币语言中被称为地址。你可以常常听到地址这个词，用于比特币或加密货币相关的内容中，而地址其实就是公钥的哈希值。作为去中心化身份管理方案的一部分，它就是某人凭空捏造的一个身份而已。

安全性与随机性

你可以不经过中央机构而生成一个身份的概念可能看起来有悖常理。毕竟，如果有人刚好就生成了跟你一样的密钥，他不就能偷走你的比特币吗？

我们给你的回答是，别人生成一个与你的相同 256 位密钥的概率如此之

小，在实践中，我们不需要担心它会发生。总而言之，我们保证这种情况绝不会发生。

一般来说，与新手的直觉不同的是，概率系统是不可预测且难以推理的，反面的常常是真的——统计学理论使得我们可以精确地量化我们感兴趣的事件的概率，并对该系统行为做出自信的推论。

但还有一个精妙之处：概率保证只有在密钥为随机产生时为真。在现实系统中，随机的生成常常是薄弱环节。如果两个用户的电脑使用同样的随机来源或者使用可预测的随机，那么理论保证不再适用。所以，在生成密钥时使用良好随机源至关重要，以确保实践保证与理论保证相符。

乍一看，去中心化身份管理可能极具匿名性及隐秘性。毕竟，你可以自己创建一个看起来很随机的身份，同时也不用告诉任何人你的真实身份是什么。但事实并不是这么简单，随着时间的推移，你创建的身份会做出一系列的声明。人们看到这些声明便知道拥有这个身份的人做出了特定的一系列行为。他们能够开始将细节联系起来，从这一系列的行为推断出你的真实身份。随着时间的推移，一个观察者可以将这些事情联系起来，并推断出这样的结论：“天，这个人的行为好像乔（Joe），可能这个人就是乔。”

换句话说，在比特币系统中，你不需要明确地注册或揭露你的真实身份，但是你的行为模式本身可能是可识别的。这就是比特币等加密货币的基本隐秘性问题，我们将会在第6章专门讨论这个问题。

1.5 两种简单的加密货币

现在，让我们从密码术过渡到加密货币。我们之前的密码术干货在这里就要开始发挥作用了，今后我们会逐渐看到各部分之间如何相互联系，也会发现哈希函数和数字签名等密码程序的意义。在本节，我们将讨论两种很简单的加

密货币。当然，我们也需要学习本书后面大量的内容，才能深刻阐释比特币本身的运作机制。

高飞币

第一个是高飞币（GoofyCoin，此币的创造者叫高飞），它应该是我们能想到的最简单的加密货币。高飞币只有两个规则，第一个规则是指定高飞可以随时创建新币，且这些新创建的币都属于他。

为创建新币，高飞生成一个他之前从未生成的唯一的货币编号（uniqueCoinID），并建立字符串“CreateCoin [uniqueCoinID]”。然后，他使用秘密签署密钥计算这个字符串的数字签名，该字符串与高飞的签名就构成一单位币。任何人都可以验证该新币包含高飞有效签名，因此该新币为有效币。

高飞币的第二个规则是，拥有此币的人可以将其转给其他人。转移一只币不是简单地将币数据结构发送给接受者，而是必须通过密码程序来完成。

假设高飞想把他创建的一只币转给爱丽丝。未达成这个目的，他需要创建一个新的声明表示“将此币支付给爱丽丝”，在此声明中“此币”就是该币的哈希指针。如上所述，身份其实就是公钥，因此“爱丽丝”指的就是爱丽丝的公钥。最后，高飞签署代表该声明的字符串。因为高飞是起初拥有该币的人，他必须签署花掉该币的任何交易。一旦由高飞签署的代表他的交易的这个数据结构存在，爱丽丝便拥有这个币。她可以向任何人证明她拥有这个币，因为她可以展示有高飞有效签名的数据结构。此外，它也指向曾经为高飞所有的一个有效币。因此，该币的有效性及所有权在系统中就不言自明了。

一旦爱丽丝拥有了这个币，她也可以花掉它。为达到这个目的，她创建了一个声明表示“将这个币付给鲍勃的公钥”，此时“这个币”就是她所有的那个币的哈希指针。当然，爱丽丝要签署该声明。任何看到这个币的人都可以验证鲍勃是其所有人。他们可以根据哈希指针链追溯到该币的创建及验证每一个步骤，这就是其合法所有人签署了一份声明表示“将这个币支付给 [新的所有人]”，详见图 1.10。

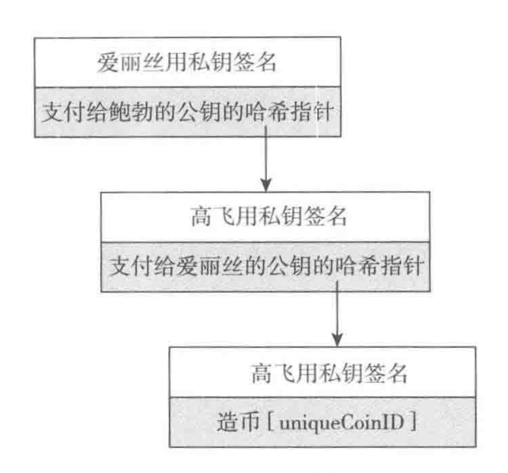


图 1.10　高飞币交易

注：该图示例了货币创造的过程和被花费过两次的过程。

总结一下，高飞币的规则是：

- 高飞可以通过签署声明表示他使用唯一的货币编号来创建一个新币。
- 币的所有人可以通过签署声明表示“将这个币转给 X”（其中 X 为公钥），将其转给另一个人。
- 任何人都可以验证一只币的有效性，跟随哈希指针追溯到它是由高飞创建，并验证过程中所有签名。

当然，高飞币有一个致命安全隐患。假设爱丽丝通过把她签署的声明发送给鲍勃，即将她的币转给鲍勃，但并没有告诉其他人。她也可以创建另一个签名，声明将同样一只币转给了查克（Chuck）。对于查克来说，这看起来是一个完全有效的交易，而他是该币的所有人。鲍勃和查克似乎都可以有效表示自己是那个币的所有人。这个就是所谓的**双重支付**（double spending）——爱丽丝将同样一只币花了两次。我们一看就知道货币是不能这样花的。

事实上，双重支付是任何加密货币需要解决的主要问题之一，高飞币没有解决这一问题，因此不安全。

高飞币很简单，其货币转移机制其实与比特币非常相似，但是因为它并不安全，因此并不适合作为加密货币。

财奴币

为解决双重支付问题，我们会涉及另外一个加密货币，我们将其称为财奴币（ScroogeCoin）。财奴币是以高飞币为基础创建的，但在数据结构方面更复杂。

第一个主要概念如下：一个叫财奴的指定实体将负责公布包含所有发生过的交易历史记录的仅增账目（append-only ledger），账目的仅增特性保证了写入这个账目的任何数据都会永久保留下来。如果账目真的为仅增，通过要求所有的交易在被接收前都写入项目，我们可以用其防止双重支付的发生。这样，如果之前币已经转给了一个不同的所有者，大家都可以看到。

为执行这个仅增功能，财奴可以建立一个区块链（我们之前已经讨论过其数据结构），对于区块链，财奴要进行数字签名，因此，这从而就形成了一系列数据块，每个数据块都包含一次交易（在实践中，一种优化的做法是将多次交易放入同一个区块中，比特币就是这样做的），每个区块包含交易的ID、交易的内容，以及上一个区块的哈希指针。财奴数字签名是针对最后一个哈希指针（它约束整个结构中所有的数据），并将签名与区块链一起发布，见图1.11。

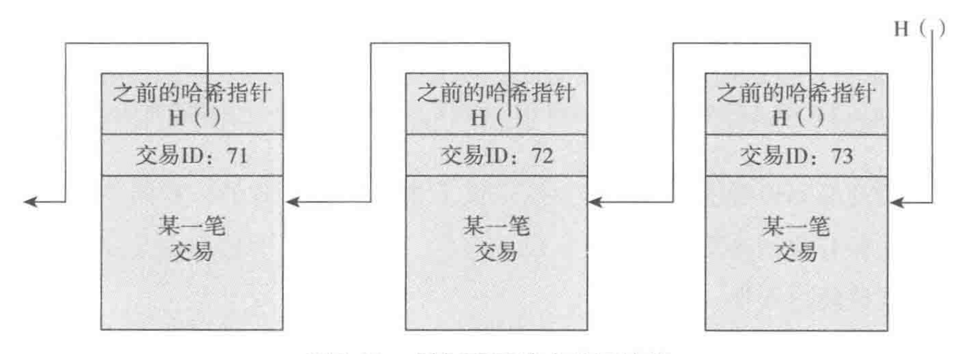


图1.11　财奴币系统中的区块链

在财奴币中，只有在由财奴签名的区块链的交易才算数。任何人都可以通过核查财奴在区块中的签名来验证交易是否经过财奴的支持，财奴会确保不会支持企图双重支付，也就是不会支持已经支付过的币的交易。

为什么除了让财奴签署每个区块，我们还需要一个带哈希指针的区块链？这样做是保证仅增特性。因为财奴有可能试图增加或移除交易记录，或者改变已有交易，而一旦有了哈希指针，将会影响到后面所有的区块。只要有人监督财奴发布的最新哈希指针，变化会很明显，并可以被轻易发现。在一个财奴分别签署不同区块的系统中，你需要记录他签署的每一个签名。采用区块链，两个不同的人可以轻易验证他们确实观察到了同样的，由财奴签署的交易记录。

财奴币中有两种交易。第一种是造币（CreateCoins），类似于在高飞币中，高飞可以创建新币的程序，而财奴将其进行了扩展，那就是可以在一次交易中创建多个币量，见图 1.12。

交易ID：73		类型：造币
被创造的货币		
序号	数量	造币记录
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

虚拟货币ID 73（0）
虚拟货币ID 73（1）
虚拟货币ID 73（2）

图 1.12　造币交易

注：造币交易创造多个货币。每一个货币在交易中都有一个序号。其次，每一个货币也有一定的数量，来对应某个数目的财奴币。最后，每一个货币还有一个造币记录，在货币被制造出来的时候对应的公钥。因此，造币交易创造了多个不同数量和归属于不同拥有者的新货币。我们将这些货币称为虚拟货币 ID，指的是该次交易中交易 ID 和货币序号的组合。

造币交易如果是由财奴签署，从定义上说它总是有效的。我们不会担心财奴什么时候有权创建新币或者可以创建多少，正如我们不担心在高飞币中，高飞可以创建新币那样。

第二种交易是付币（PayCoins）。这一交易会消耗币，就是说消除它们，并创建具有相同总值的新币。新币可能属于不同的人（公钥），这一交易必须由每一个支付该币的人来进行签署。因此，如果你是本次交易中将会消耗的某只币的所有人，那么你就需要数字签署该交易，表明你同意花掉这只币。

财奴币的规则阐明，如果以下四个条件为真，付币交易有效：

- 被消耗的币为有效货币，即它们是在之前的交易中创建的。
- 被消耗的币没有在之前的某交易中被消耗掉。就是说，本次交易不是双重支出。
- 本次交易产生的币值量等于消耗的币值量，也就是说，只有财奴才可以创建新币。
- 本次交易被消耗的所有币均有其所有者的有效签署。

<table>
<tr><td colspan="2">交易 ID：73</td><td>类型：付币</td></tr>
<tr><td colspan="3">消耗的虚拟货币 ID：
68（1），42（0），72（3）</td></tr>
<tr><td colspan="3">被创造的货币</td></tr>
<tr><td>序号</td><td>数量</td><td>造币记录</td></tr>
<tr><td>0</td><td>3.2</td><td>0x...</td></tr>
<tr><td>1</td><td>1.4</td><td>0x...</td></tr>
<tr><td>2</td><td>7.1</td><td>0x...</td></tr>
<tr><td colspan="3">签名</td></tr>
</table>

图 1.13　付币交易

如果所有条件都满足，那么付币交易有效，并且财奴会接受交易（见图 1.13）。他会通过将其附加到区块链上，将交易写入历史记录。之后，每个人都可以看见交易发生了。只有在这时，参与者才可以接受交易实际发生了。直至发布之前，它都可能是一个被双重支付抢占的交易，即使前三个条件都被满足。

这个系统中的货币是不可变的——它们不会被改变、细分或者联合。每个币都在一次交易中被创建一次，然后在之后的其他交易中被消耗。但是我们可以通过交易对货币进行细分或联合，来实现相同的效果。例如，为了细分一只币，爱丽丝创建了消耗该币的新交易，然后生成了两个具有同样总值的新币，这两个新币可以再次分配给她。因此，虽然在本系统中币是不可变的，但是它具有除了可变币以外的系统的所有灵活性。

现在，我们来看一下财奴币的核心问题，财奴币的工作原理是人们可以看见哪些币是有效的。它防止双重支付，因为每个人都可以查看区块链，看到所

有交易都是有效的，每一只币确实都只被消耗了一次，但其问题是，财奴的权利太大了。他虽然不能创建虚假交易，因为他无法伪造其他人的签名，但是他可以停止支持其他用户的交易，不为他们提供服务并让他们的货币无处可花。如果财奴是贪婪的（正如与他同名的卡通形象一样），他可以拒绝公开交易，除非其他人向他支付强制性交易费。当然了，财奴还可以想要多少币，就给他自己创建多少。或者财奴也可能厌倦整个系统，因此完全停止更新区块链。

这里的问题就是中心化。虽然财奴本身满意这个系统，我们用户可能会不满意。财奴币虽然看似是一个不切实际的方案，但是在很多早期的密码系统研究中，确实假设过一些中央可信机构，还特别被称为银行。毕竟，绝大多数现实世界货币的确有可信发行人（通常为政府造币厂）负责创建货币，并决定哪些钱币为有效货币。但是，具有中央机构的加密货币纷纷在实践中失败。原因有很多，回头来看，我们似乎很难让人们接受有中央机构的加密货币这个事物。

因此，为改善财奴币，并建立一个可行系统，我们需要解决的主要技术问题是：我们是否能让系统“去财奴化”？也就是说，我们是否能放弃中心化的财奴人物？我们能够有一个在很多方面像财奴币一样运作的加密货币，但没有中央信任机构吗？

为回答这些问题，我们需要解决所有用户如何在交易历史记录发生后，一致同意采用一个公开区块链，他们必须一致同意哪些交易有效、哪些交易是实际发生了。他们还需要能够用一种去中心化的方式分配 ID。最后，新币的铸造也需要通过去中心化的方式进行掌控。如果我们可以解决所有这些问题，那么我们可以创建一个如同财奴币那样的货币，但确实没有中心化的机构。实际上，这样的一个系统就与比特币非常相像了。

延伸阅读

史蒂芬·列维（Steven Levy）的《密码术》，从一个令人愉悦的、非技术的角度看待现代密码术的发展，及其背后的人和事：

Levy, Steven. *Crypto: How the Code Rebels Beat the Government—Saving Privacy*

in the Digital Age. London：Penguin，2001.

现在密码术还是一个较为理论化的领域，密码学者使用数学以一种较为正规的方式定义其基础知识、协议以及其他被用户期望的安全特性，并根据关于特定数学问题的计算复杂性中被广泛接受的假设，来证明它们的安全性。本章我们使用到了直觉语言来讨论哈希函数及数字签名。对于有兴趣用更为严格的数学的方式，以及想更深入探索这些概念及其他密码学理论的读者，我们推荐你阅读：

Katz，Jonathan，and Yehuda Lindell. *Introduction to Modern Cryptography*，second edition. Boca Raton，FL：CRC Press，2014.

对于应用密码学概述，参见：

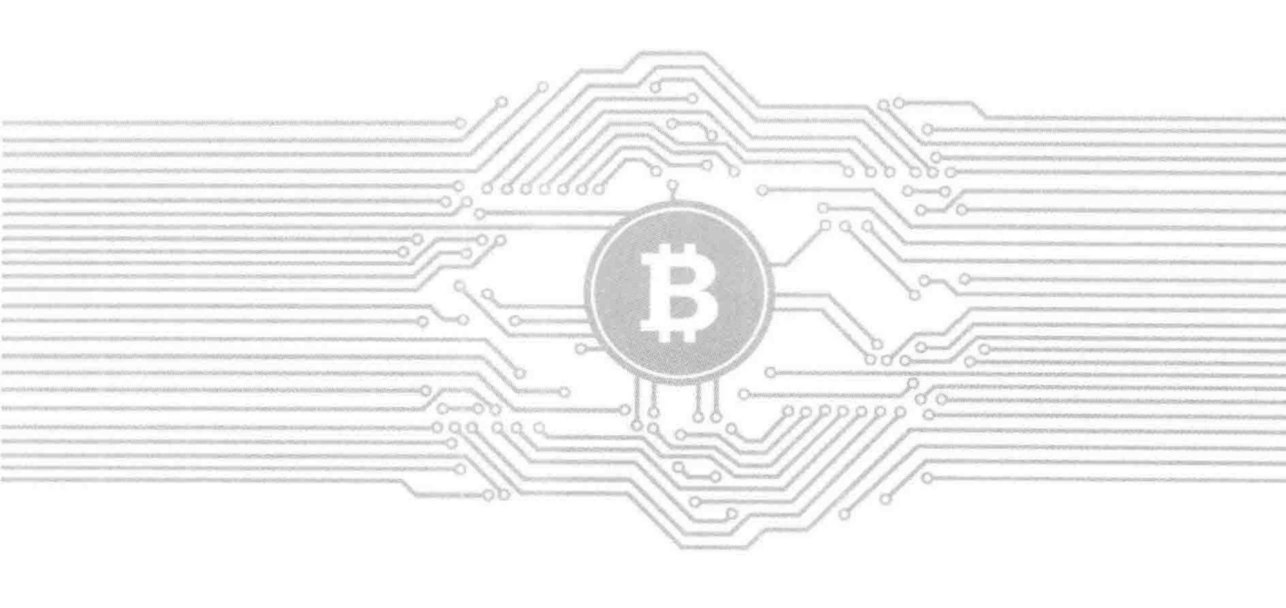
Ferguson，Niels，Bruce Schneier，and Tadayoshi Kohno. *Cryptography Engineering：Design Principles and Practical Applications*. Hoboken，NJ：John Wiley & Sons，2012.

精读定义 SHA－256 的 NIST 标准是了解密码学标准的有效方式：

NIST. "Secure Hash Standards，Federal Information Processing Standards Publication." FIPS PUB 180-4. Information Technology Laboratory，NIST，Gaithersburg，MD，2008.

最后，请参考讨论 ECDSA 签名算法标准化版本的论文：

Johnson，Don，Alfred Menezes，and Scott Vanstone. "The Elliptic Curve Digital Signature Algorithm（ECDSA）." *International Journal of Information Security* 1（1）2001：36－63.



第2章

比特币如何做到去中心化

在这一章，我们将讨论比特币如何做到去中心化。在第1章中，我们讨论了比特币底层加密算法的基础，最后我们谈到了财奴币。作为一种以账本为基础的记账式加密数字货币，财奴系统已经做得确实不错了，但它有一个很突出的问题，那就是该系统非常依赖一个被称为“财奴”的中心化权威。在第1章的最后，我们提出了财奴币去中心化的问题，或者说如何去财奴化。在本章，我们将着重讨论这个问题。

通读完本章，我们将注意到比特币并不是完全使用纯技术手段，而是将技术手段与一种明智的激励机制相结合，做到了去中心化。本章的最终目的会使你对去中心化有一个通盘的认识，同时也对比特币运作机制有所了解，并且懂得为什么比特币确实是安全的。

2.1 中心化与去中心化

去中心化是一个重要概念，这个概念并不是比特币独有的特性。在各种数字技术领域，中心化与去中心化两派的竞争也越来越多见。为了更好地理解竞争模式在比特币里的表现，我们有必要了解一下两派竞争在其他不同技术领域的竞争焦点。

互联网其实就是一个著名的去中心化系统。但在早期，互联网是在与美

国在线（American On-Line，简称 AOL）以及 CompuServe[①] 等围墙花园式信息服务体系的竞争中，逐步变得越来越风行。电子邮件的实质也是一种简单邮件传输协议（Simple Mail Transfer Protocol，简称 SMTP）的去中心化系统。尽管电子邮件也受到像脸书（Facebook）、领英（LinkedIn）这些中心化私有信息系统邮箱服务体系的挑战，但电子邮件仍然是人与人之间进行通信的一种被默认的选择。其实，我们已经不能简单将像即时短信或者短信等通信手段归类为是中心化，或者是去中心化模式，这些通信方式往往是一种混合模式。在社交网络中，尽管有很多爱好者、技术开发人员，甚至还有企业者也在尝试用去中心化的方式来替代像脸书、领英这样的中心化系统，但目前这些中心化系统仍具统治地位。事实上，中心化与去中心化的竞争在数字时代之前就已经存在，在电话、无线电、电视及电影的发展史上，我们都曾看到过这两种模式的竞争。

中心化与去中心化也并非水火不容，其实没有一个系统是完全中心化，或者是完全去中心化的。比如，电子邮件其实是一个去中心化系统，它基于一个标准的中心化协议 SMTP，任何人只要愿意，都可以设计一个自己的电子邮件服务器。但实际情况是，只有一小部分电子邮件服务商在这个领域占据着统治地位。类似，虽然比特币系统是去中心化的，但比特币交易所（将比特币转换成其他货币的平台）、钱包软件以及用户管理比特币的软件，可以是中心化的，也可以是去中心化的。

有了以上的考虑，我们把比特币如何做到去中心化这个问题拆解为下面五个问题：

1. 谁在维护交易账本？

---

① CompuServe，美国最大的在线信息服务机构之一。CompuServe 产品于 1979 年问世，它提供留言板、新闻和信息、电子商务以及其他类似网络功能的服务。这款产品的问世时间远远早于网络。美国在线在 20 世纪 90 年代早期的崛起，使得 CompuServe 退居美国第二大在线服务商。不久之后，CompuServe 不得不同互联网进行竞争，它变成了一个不那么令人满意的互联网服务提供商。而且，随着用户更多地使用互联网，CompuServe 风光一时的留言板也开始被人抛弃。1997 年，美国在线收购了 CompuServe。正如网景一样，CompuServe 成为美国在线用在其他产品上的标示。现在，CompuServe 只是一个半门户网站。——译者注

2. 谁有权利批准哪个交易是正当有效的?
3. 谁在制造新的比特币?
4. 谁在制定系统变化规则?
5. 比特币是如何取得交易价值的?

前三个问题反映了比特币协议的技术细节，我们将在本章重点讨论。

比特币系统的不同方面是从不同点涉及了中心化及去中心化。点对点网络是最接近去中心化的体系，任何一个人都可以运行一个比特币节点，而且基本没有什么入门门槛，用户只需要上网下载一个比特币客户端，就可以在其个人电脑上运行一个节点，现在全球有成千上万个这样的节点。在本章 2.4 节中我们将要学习比特币挖矿（bitcoin mining)，从技术上讲，挖矿过程也是向所有人开放的，但挖矿需要很多资金投入。正因为如此，挖矿领域具有非常高的中心化及挖矿能力集中的倾向。比特币社区里有许多人认为这种现象并不可取。第三点是关于比特币运行节点软件的更新，这涉及何时以及如何更新系统规则。大家可以想象，就像电子邮件系统那样，这些节点可能有各种根据相同方式但通过不同手段实现的不同版本。但在实际上，绝大多数节点用的都是社区里被大家公认的有权威的资深开发者开发出来的软件。

2.2 分布式共识

在前一节，我们笼统地讨论了去中心化和中心化。现在我们从一个更为技术性的层面看一下比特币的去中心化。接下来，我们会遇到一个被称作“共识”（consensus）的重要概念，特别地，还有“分布式共识”（distributed consensus)。建立一个分布式的电子现金系统的关键技术问题，就在于要达成分布式共识。直观地说，你可以想象我们的目标就是要将第 1 章提到的财奴币去中心化。

分布式共识有各种应用，计算机界对其也研究了多年，传统具有启发式的应用就是提高分布式系统的可靠性。设想你在管理一个社交网络公司的后端平

台，比如微信，像这样庞大的系统通常有几千台甚至几万台服务器，这些服务器组成了一个巨大的分布式数据库，数据库中记录了这个系统里发生的各种活动，而每条信息都会被记录在后端的若干个节点上，对于整个系统的状态，这些节点必须要做到同步。

分布式共识协议的意义远远超出了传统意义的范畴。一旦具备了这样的体系，我们就可以建立一个庞大的分布式**键值**（key-value）存储库，该类存储库可以将任意数据如身高、名字等对应一个相应的开启键，基于此，许多应用得以实现。例如，我们可以建立一个分布式域名系统，将人脑易于理解的域名与IP地址进行配对，我们也可以建立一个公钥目录，这个目录可以把公钥与电邮地址（或者其他真实世界中的身份证明）对应起来。

以上讨论在直觉上说明了分布式共识的大概含义。对于分布式共识，我们还是要给出一个技术定义，以此我们可以判别一个协定是否符合分布式共识的要求。

> **分布式共识协议**　在一个有 n 个节点的系统中，每一个节点都有一个输入值，其中一些节点具有故障，甚至是恶意的。一个分布式共识协议有以下两个属性：
>
> - 输入值的中止须经所有诚实节点来确定。
> - 这个输入值必须由诚实节点来生成。

那么以上概念在比特币里又是什么含义呢？想要理解分布式共识在比特币中的用途，我们需要记住比特币是个点对点的系统。当爱丽丝向鲍勃付款的时候，她其实是在向构成比特币网络上的所有节点广播其交易行为，见图 2. 1。

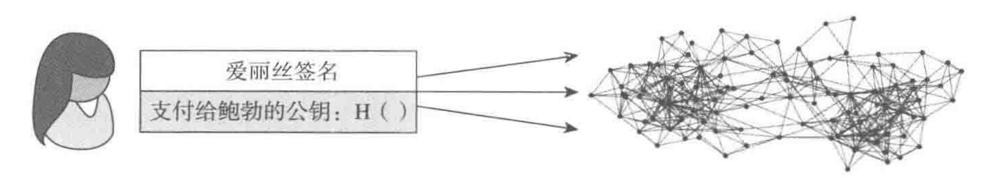


图 2. 1　广播交易

注：为了向鲍勃付款，爱丽丝需要向整个比特币点对点网络进行广播。

顺便提一下，你可能注意到，当爱丽丝向整个比特币点对点系统广播时，鲍勃的计算机并不一定在图 2.1 的网络中。当然鲍勃也有可能在这个网络上运行着一个节点，如果鲍勃想在爱丽丝转币给他时及时被系统通知，运行一个节点当然是个好主意，但其实这并不重要，鲍勃是否运行节点并不影响他收到爱丽丝转给他的比特币。

在比特币网络里，节点到底要达成什么样的共识呢？网络里有各种各样的用户在向网络广播交易，节点必须对哪些交易可以进行广播和交易发生的次序达成共识，以此系统将形成一个唯一的全球交易总账。回想我们在第 1 章 1.5 节中曾提到的财奴币将交易打包成块，对信息进行优化处理。类似地，在比特币体系里，我们也将每个区块进行共识处理。

在任何时点，所有在点对点网络上的节点都有包含一系列区块的总账本，每个区块中都包含了已经被所有节点达成共识的交易清单。除此之外，每个节点还有一堆没有被打包进入区块的交易，就是那些网络节点已经被通知、交易已经发生，但还没有被写进区块的交易。网络节点对于这些交易还没有达成共识，所以每个节点都有一个略有差异、尚待确认的交易池。在实际中，点对点网络是不完美的，所以有些节点听到了交易，而有些节点却没有听到。

那么，所有的节点是如何对一个区块达成共识的呢？一个方法是，在一个时间段里，比如说每隔十分钟，每个节点都提议，自己的未被认可的交易成为已经达成共识的区块链后面的下一个区块，然后那些节点会执行一些共识协议，每个节点把自己提议的区块作为输入。但不可避免地，有些节点可能是恶意的，存心要把不当交易放进区块里，其他节点则是诚实的。如果共识协议能够顺利完成，一个正当有效的区块会被选作输出值。尽管有些被选出的区块是由一个节点提交，但只要这个区块是正当有效的，输出就是正当有效的。这时候可能有人会指出，这个被选出的区块可能未包含所有的正当有效的交易，但这并没有关系，如果有些正当有效的交易没被放进区块，它们可以等待下一次机会。

前面所谈到的这个办法与比特币系统有些相似之处了，但实质还是不完全

一样。以上做法有几个技术上的问题：第一，达成共识一般是个难题，因为有些节点会死机或是根本就是恶意节点；第二，就比特币而言，点对点网络是不完美的，并非所有对应的节点是两两相连的，互联网链接的不良可能会造成网络问题，要执行一个所有节点都参与的共识协议好像并不现实；第三，由于交易信息是分布在整个互联网上，信息传递会有严重延迟。

延迟与全球时间

比特币协议达成共识时必须直面两大障碍：其一是不完美网络，例如信息延迟和节点死机，其二是某些故意搞破坏的节点。

严重网络延迟导致的一个后果是，节点之间没有一个统一的全球时间概念。意思是，并非所有节点都能根据每个交易的时间戳来达成交易时间共识，因此，共识协议不能执行以下指令："在第一步里发了第一个消息的节点必须在第二步里执行 X。"这一做法根本无法执行，因为所有的节点对于谁在第一步中发出第一个信息有不同的看法。

不可能性结论

在全球时间上的不统一，给共识协议算法带来了很多限制。事实上，由于这些限制，许多关于分布式共识的文献都对是否能达成共识持悲观态度，有许多达成共识具备不可能性的结论已经被证实。一个经典案例就是"拜占庭将军问题"（Byzantine Generals Problem），这个经典难题是这样阐述的：拜占庭是东罗马帝国的首都，它的军队分成多个师，每个师都由一个将军统领。这些将军通过信使进行交流，来达成一个共同作战方案，有些将军可能是叛徒，想故意破坏这个过程，这会造成那些忠诚的将军也无法达成一个统一的作战计划。解决这个难题的办法就是让那些忠诚的将军在这样的情况下达成统一作战方案，而避免那些叛徒对作战方案的误导。事实证明，如果叛徒数量超过 1/3 时，这个难题将无法克服，那些忠臣的计划终会被叛徒们破坏。

还有一个更为微妙的关于不可能性的结论，这就是著名的“Fischer-Lynch-Paterson 不可能结果”[①]，该名称以最初的作者而命名。该结果指出，在一定的条件下（包括节点行为具有确定性特征），甚至在只有一个缺陷的过程中，达成共识都是不可能的。

尽管有这些“不可能性结论”，还是有文献谈到了一些共识协议。比较著名的就是 Paxos 算法协议。Paxos 算法做了一些妥协，一方面，Paxos 算法能做到不产生不一致的结果；另一方面，Paxos 算法所做的妥协是，在一定条件下（虽然是不常见的情形），该协议会死机卡住，从而无法继续运行。

打破传统上的假设

但好消息是，这些所谓的“不可能性结论”都是在一些特定的模式下才成立，这些结论是针对分布式数据库的研究，这些模型不能完全套用到比特币身上来，比特币本身就打破了很多原来分布式数据库所做的假设。这些结论其实从某一方面让我们更明白了那些特定模式，由此或许可以真正对分布式共识给出解决方案。

具有讽刺意义的是，就目前对共识的研究来说，比特币实际运行情况下远比理论上告诉我们的要好得多，这就是比特币让专家们跌破眼镜之处。我们看到分布式共识在比特币里运行良好，但我们还没有建立理论来充分解释为什么会这样，但无论如何，完善理论对将来的发展还是十分重要的，理论结果可以使我们预测，甚至预防未来可能的攻击和问题。我们一旦具备了较强的理论依据，来解释比特币分布式共识的良好运作机制，我们才能真正地对比特币的安全性和稳定性做出保证。

比特币到底打破了经典模型里的哪些假设呢？第一，比特币引进了奖励的理念，这对分布式共识协议来说是一个全新的理念，这也只有在比特币里才可

① Fischer-Lynch-Paterson 不可能结果，是 Michael J. Fischer、Nancy A. Lynch 和 Michael S. Paterson 在论文 *Impossibility of distributed consensus with one faulty process* 中证明的一个结论，称得上是分布式理论中最为深刻的结论，大致表述如下：“在一个多进程异步系统中，只要有一个进程不可靠，那么就不存在一个协议，此协议能保证有限时间内使所有进程达成一致。”——译者注

能实现，因为比特币也是个货币，所以人们自然而然地会为了金钱奖励而变得诚实起来。所以，比特币并没有真正解决分布式共识问题，它只是在特定货币系统下解决了这个问题而已。

第二，比特币体系包含随机性这个概念。在后面两节里我们将会看到，比特币的共识算法很大程度上依赖于随机性。此外，它也不再纠结于规定共识的起点与终点。相反，共识是通过一段较长的时间而达成的，在实际系统中，达成共识大约需要一个小时左右。但即使在一个小时以后，节点们也无法确定哪一个交易块应该进入总账本。但随着时间的流逝，我们对某一个块的认识与最终总体共识相吻合的概率将越来越大，观点出现分歧的概率按指数级下降。比特币在以上方面的不同，让它能够逾越传统理论关于分布式共识不可达成这一鸿沟。

2.3　使用区块链达成没有身份的共识

在这一节里，我们将探讨比特币共识算法的技术细节。回忆一下，我们在前面曾说过，比特币中的每个节点并没有一个稳定的、长期的身份，这一点也是与传统分布式共识算法的不同之处。身份缺失的原因是，在一个点对点网络中，没有一个中央权威机构来发放身份，并保证它们没有制造节点。用技术术语来说，乱造节点就是所谓的“女巫攻击”（sybil attack）现象。女巫就是恶意黑客制造的不同节点，这些节点看起来像是对应不同的身份的人，其实是由一个人在幕后控制。另一个原因是化名制（pseudonymity），也是比特币想达到的一个目标，所以即使可以替所有节点建立唯一真实身份，我们也不想那样做。虽然比特币还是不能保证真正的匿名，一个用户用不同身份做的不同交易还是有办法被最终追踪到，但比特币的特性毕竟没有强迫大家用真实身份来加入。这是比特币的重要特性，也是比特币系统的核心理念。

如果所有节点都有真实身份的话，那么设计上会更加容易。有了真实身份，我们就能够以这样的方式发出协议指令，比如“编号最小的节点开始做某些动

作”，在没有真实身份前提下，系统能设计的指令就受到很多限制，但设计真实身份最主要的考虑是安全上的便利。如果节点的身份可以被识别，就不能随便地制造新的节点身份出来。那样的话，我们就可以假设有恶意节点的数量，然后部署安全措施来防范。基于以上原因，缺少真实身份给比特币的共识协议带来很多难点。

我们可以做一个较弱的理论假设来弥补这个先天的不足。假设我们可以在系统里随意选一个节点，一个比较好的比喻是——就如同在彩票站，或是在任何一个难以辨别每个人身份的系统中，我们给每一位顾客发出彩票或是一个识别牌，之后我们就可以开始抽奖，与奖号对应的人就会中奖。现在我们想象一下在比特币的世界里，我们假设也可以做到这一点。我们再假设，这个彩票的印制过程与发放办法是足够聪明的，如果一个黑客想制造出许多女巫节点来，最后所有这些节点也只能拿到一张彩票。也就是说，这个黑客无法通过制造假的节点来增强他的力量。如果你觉得我们做的假设太多了，请不要担心，我们在以后会消除这些假设，并在后文会详细说明，在比特币系统中，与这些假设相对应的性质是如何实现的。

隐性共识

对随意节点选择的假设可以让“隐性共识”（inplicit consensus）成为可能。我们的共识协议有多个回合，每个回合都对应着区块链里的一个块。在每一个回合里，一个随机节点会被选中，然后这个节点可以提议这个链的下一个区块。这时没有共识算法，也没有任何投票过程来决定哪个区块会被选中，随机被选中的节点会直接决定区块链的下一个区块，但万一这个节点是恶意的呢？针对这个问题，还是有应对办法的，解决方法就是隐性共识。其他节点可以通过隐性地接受或是拒绝前面这个被随机选择出来的节点。如果接受，它们会在这个块之后接龙下去；如果拒绝，它们忽略这个新的区块，而是选择前一曾经接受的区块，来继续接龙下去。大家还记得，每一块都记录着前一块的哈希值。这就是节点选择在哪一块来继续接龙的技术处理方式：比特币共识算法（简化版）。

这个算法的简化假设是，可以随意选择一个节点，这些节点都不会受到女巫攻击的影响。

1. 新的交易被广播到所有节点上。
2. 每个节点都将新的交易放进一个区块。
3. 在每个回合，一个随机的节点可以广播它的区块。
4. 其他节点可以选择接受这个区块，前提是如果区块里的交易都是正当的（有真的签名）。
5. 节点们可以把以上区块的哈希值放进自己的区块里，以此来表示它们对那个新区块的认可。

我们现在一起来研究一下为什么这样一个共识算法是有效的。为此，我们假设有一个叫爱丽丝的黑客，她想要破坏这个共识过程。

窃取比特币

爱丽丝能够窃取属于另一个用户，不受她控制的地址里的比特币吗？答案是否定的。即使这一轮是由爱丽丝提议区块链上的下一个区块，她也不可能窃取别人的比特币。这么做的话，爱丽丝需要发起一笔有效的交易来花掉这个比特币。这就要求爱丽丝伪造比特币拥有者的签名，然而如果数字签名机制是安全的，她是无法办到的。只要背后的密码学基础是牢靠的，她就无法轻易窃取比特币。

拒绝服务攻击

让我们来考虑另一种攻击。假设爱丽丝不喜欢叫鲍勃的某个用户，爱丽丝可以决定她不把鲍勃发起的任何交易放进她所提议的区块里。换言之，她拒绝提供服务给鲍勃。尽管这是爱丽丝可以开展的有效的攻击，但幸好这不过是个小问题。如果鲍勃的交易没有被放进爱丽丝所提议的下一个区块，鲍勃只要等到下一个诚实节点发起区块的时候，他的交易记录就会被放进这个区块里。所以这其实也不算是一个有效的攻击。

双重支付攻击

爱丽丝也可能会发起一个双重支付攻击。要理解爱丽丝如何发起这种攻击，我们可以假设爱丽丝是鲍勃开的网店或网站的一名顾客。鲍勃提供一些比特币付费的在线服务，比如软件下载。双重支付攻击是这样的：爱丽丝在鲍勃的网站选中一件商品并加入购物车中，此时服务器要求付款。然后，爱丽丝在她的地址上向鲍勃的地址发起了一笔比特币交易，并向整个网络广播这笔交易。我们假设由某个诚实节点来制造下一个区块，并把这笔交易放进这个区块中。因此，现在就有了一个由诚实节点发起，包含代表爱丽丝向商家鲍勃支付这笔交易在内的区块了。

我们还记得一个交易就是一个数据结构，里面有爱丽丝的数字签名，一个付给鲍勃的公钥（地址）的指令和一个哈希值。这个哈希值代表了一个指针，指向先前的一笔交易的输出，即爱丽丝之前收到并于现在消费比特币。这个指针必须指向一个已被共识链上的某个之前的区块所认可的交易。

顺便说一下，有两种容易混淆的不同类型的哈希指针。一种是在区块内用来表示接在之前哪个区块后面的哈希指针；另一种是在交易里的一个或多个，用来指向之前交易里说明比特币来源的哈希指针。

我们回到爱丽丝如何发起双重支付攻击这个问题。最新的一个区块由一个诚实节点产生，其中包含爱丽丝下载软件向鲍勃付费的交易记录。当看到这笔交易被放入区块链后，鲍勃认为爱丽丝已经向他付款，便允许爱丽丝下载软件。假设在下一个回合被随机选中的节点恰巧被爱丽丝所控制。现在因为爱丽丝可以提议下一个区块，她可以选择忽略掉前面那个包含她支付给鲍勃的那笔交易的区块，而产生一个包含指向之前区块指针的区块。不仅这样，在这个区块里，爱丽丝可以放进一笔交易，把她付给鲍勃的币转到一个被她所控制地址里去。这就是一个经典的双重支付攻击。因为这两个交易用的是同一个币，只有一个交易可以被放进区块链。所以如果爱丽丝成功地把币转到她控制的地址，那个她付币给鲍勃的交易记录将变得无效，因为它将不会被放进区块链里。这一过程详见图2.2。

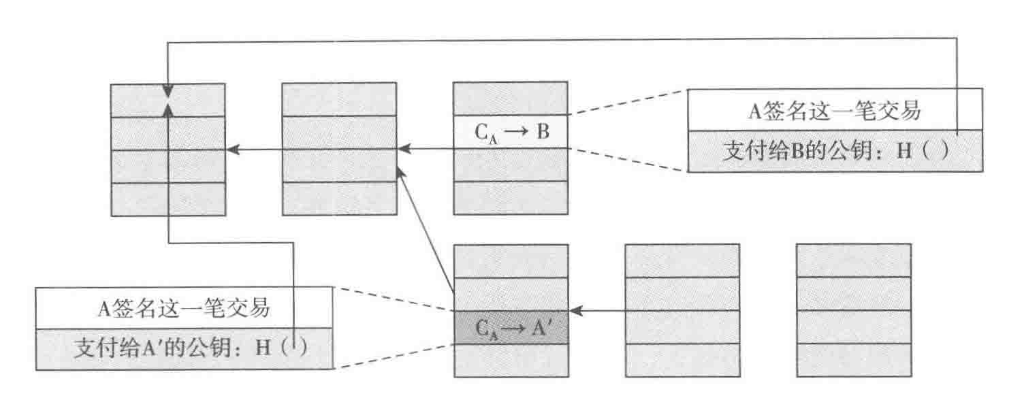


图 2.2　双重支付攻击

注：爱丽丝创建了两笔交易：一笔是她付给鲍勃比特币的交易，另一笔是她将这笔比特币重复支付到她控制的另一个地址。因为这两笔交易用相同的比特币支付，所以只有一笔会被放进区块链。图中的箭头表示一个区块链接到前一个区块的指针，通过在前一个区块自己的内容中包含了一个哈希值进行了扩展。C_A代表爱丽丝拥有的币。

那我们如何知道这个双重支付攻击是否能成功呢？这取决于最后哪个区块会被纳入长期的共识链，是爱丽丝转给鲍勃的区块，还是爱丽丝转给爱丽丝自己的区块。是什么决定了哪一个区块被纳入呢？诚实节点会遵守在最长有效分支后面延展这一规则，那到底在哪个分支后面延展呢？并没有明确的答案！目前来看，这两条分支长度一样，它们的区别是仅在于最后一个区块，并且这两个区块都是有效的。选择下一个区块的节点可以决定建立在其中一个区块上。这个选择就决定了双重支付攻击的成功与否。

微妙之处在于：从道德角度考虑，这两个分支截然不同，一个是包含付给鲍勃交易的区块，一个是包含爱丽丝把这些币双重支付给她自己地址交易的区块。但这个区别仅仅建立在我们知道爱丽丝先支付给鲍勃再试图双重支付这个故事的基础上。但从技术角度来看，这两笔交易完全一致，且都有效。节点没有办法分辨出哪一个是道义上合理合法的交易。

实践中，节点往往用延展它们在点对点网络里最早听到的区块这种启发式的方法。但这不是一个无懈可击的法则。在一些情况下，因为网络延迟，很可能它们先听到的区块实际上是后被创造出来的。所以下一个提议节点至少是有可能选择在那个包含双重支付的区块上延展。爱丽丝甚至还可以贿赂下一个提

议节点来加大这个可能性。不管出于什么原因，如果下一个节点真的接受了这个双重支付的区块，那么这条链将比包含支付给鲍勃交易的那条链更长。基于此，下一个诚实节点就有可能去延展这条链，因为它更长。随着这个过程继续，这条包含双重支付的链会更有可能成为长期共识链的一部分。相反，那个包含爱丽丝支付给鲍勃交易区块的链会被网络完全遗忘，成为一个**孤块**（orphan block）。

我们现在从商家鲍勃的立场重新考虑整个情况。理解鲍勃如何保护自己不受双重支付攻击是理解比特币安全措施的重要的一部分。当爱丽丝广播她向鲍勃支付的交易时，鲍勃也在网上听着，鲍勃在下一区块被创建之前就能听到这笔交易。如果鲍勃比我们前面描述的更加草率的话，他可以在网上完成检查程序，并允许爱丽丝此时下载软件。这叫作**零验证交易**（zero confirmation transaction）。这将导致一个比前面所说的更加基础的双重支付攻击。前面所述情况，为了实现双重支付攻击，我们需要假设一个恶意黑客控制了发起下一个区块的节点。但如果鲍勃允许爱丽丝在没有收到区块链一条确认信息的情况下就下载软件，那么爱丽丝可以立刻广播一条双重支付交易，一个诚实节点就有可能把这个交易放进下一区块，而不是支付给鲍勃的那笔交易。见图 2. 3。

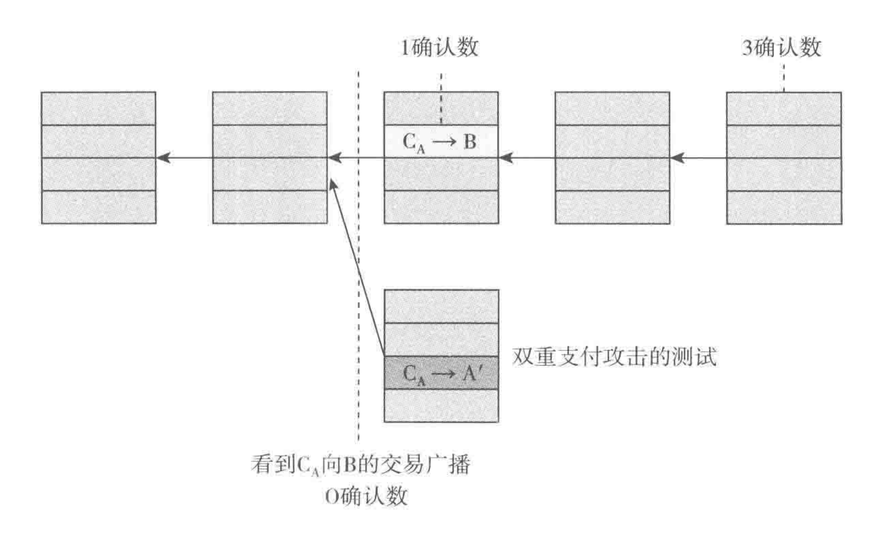


图 2. 3　从鲍勃立场来看双重支付

注：这是一个从商家鲍勃的立场来看爱丽丝做的双重支付尝试。为了保护自己免受双重支付攻击，鲍勃应当等爱丽丝向他支付的交易被区块链包含进去，并且多等几次确认。

另一方面，一个谨慎的商家甚至在看到交易被包含在一个区块后仍然不会允许爱丽丝下载软件，而是继续等待。如果鲍勃看到爱丽丝成功发起了双重支付的攻击，他会意识到那个含有爱丽丝向他支付的交易的区块有可能已经被丢弃。他应该放弃这个交易，不让爱丽丝下载软件。如果在尝试双重支付的情况下，恰巧下几个节点还是建立在爱丽丝向鲍勃支付交易的区块上，那鲍勃就相信这笔交易会被纳入长期共识链。

总而言之，一个交易得到的确认越多，它被纳入长期共识链的概率就越大。如前文所述，诚实节点总是选择延展最长的共识链。因为长链增长更多，那条含有双重支付的短链追上长链的概率会变得越来越小。在只有一小撮恶意节点的情况下，这个结论尤其正确。因为短链要想赶上，这些恶意节点需要被一直连续选中。

事实证明，双重支付攻击成功的概率将随着确认的数目的增加而指数级降低。所以，如果你感兴趣的交易已经收到 k 个确认，双重支付攻击交易被纳入长期共识链的概率以关于 k 的一个函数指数级下降。在比特币生态系统里，最常见的方法是等 6 个确认。并不是 6 这个数字有什么特殊意义，只不过，这样在你等待的时间与确认你所感兴趣的交易被纳入长期共识链之间做了很好的平衡①。

总结起来，防止不正当交易完全是用密码学的方法。但这些方法被共识所加强，意思是一个节点如果想放进一个密码学上不正当的交易，这个交易不会被纳入长期共识链的唯一原因是绝大多数的节点是诚实的，不会把一个不正当交易放进区块链。另外，防止双重支付攻击完全依赖于共识，密码学不起任何作用。从密码角度来看，这两个交易都是正当有效的。但共识可以能决定哪个被放进长期共识链。最后，你无法百分之百保证你感兴趣的交易被放进了长期共识链。但指数级概率保证了不错的结果，6 笔交易过后，实质上你没有犯错的可能了。

① 如后文总结时所说，并没有一个固定的数字，但 6 是个常用的数目。——译者注

2.4 奖励机制与工作量证明

在前面的章节里，我们简单了解了比特币的共识算法，以及为什么我们直觉上相信它是安全的。但我们回想在本章一开始谈到的，比特币的去中心化一部分是通过技术手段，另一部分是通过聪明的激励设计来实现的。截至目前，我们主要关注的还是技术手段。现在，我们来讨论比特币的这个激励设计。

之前我们试图大胆相信这样的假设，在我们随机选取节点时，至少有50%的可能会选中诚实节点，这或许是有问题的。如果对颠覆这个过程的参与者有金钱奖励，这个关于诚实的假设就格外成问题，这种情况下我们无法真的假设某个节点是诚实的。所以这个问题变成了：我们是否可以给予表现诚实的节点奖励?

我们再思考下一个确认以后的双重支付尝试（见图2.3）。我们是否可以惩罚那个创建包含双重支付区块的节点？好吧，其实不行。就像我们前面说的，因为我们无法判断哪笔交易是道义上合法的。即使我们知道，我们也很难惩罚它们，因为节点没有身份。那我们反过来思考，我们是否可以奖励那些创造的区块最终被放入长期共识区块的节点？然而，同样因为这些节点并没有透露它们真实世界中的身份，我们不可能给他们的家庭地址寄去现金。要是有某种可以代替的数字货币……你大概猜到该怎么做的。我们可以用比特币来奖励创造这些区块的节点。

让我们暂停一下。之前，我们讨论的都是用抽象的算法来实现分布式共识，并不是针对某个具体的应用。我们现在要跳出模型，使用事实，我们建立这个分布式共识过程的应用实际上就是一种货币。明确地说，我们要以这种货币为单位奖励那些表现诚实的节点。

区块奖励

这是怎么做到的？比特币里有两种不同的奖励机制。其中一个就是区块奖

励。根据比特币的规则，创建区块的节点可以在这个区块中加入一笔特别的交易。这笔交易就是一个造币的交易，类似于财奴币里面的造币，节点可以指定这笔交易的接收地址。当然，节点通常都会选择一个属于自己的地址。你可以把这视为对节点在共识链上进行创建区块服务的报酬。

在写本书时，区块奖励金额定在 25 个比特币。但每生成 210 000 个区块，金额就会减半。根据区块生成的速度，我们可以看到，这个金额大概每 4 年减半一次。我们现在处在第二个 4 年。比特币存在的最初 4 年，区块奖励金额为 50 个比特币，现在是 25 个比特币。然后会不断减半。这将造成一些有意思的结果，我们不久会看到。

你可能会问为什么区块奖励能做到鼓励诚实行为。给予我们目前讨论的，从表面上看，这个节点无论提议一个正当有效的区块还是恶意伪造，都会受到奖励。但其实并非如此！想一想这个节点是如何收取奖励的？奖励只有当区块最终被纳入长期共识链才会实现。因为造币交易和其他每一笔交易一样，只有当它最终被纳入共识链，才会被其他节点接受。这就是比特币奖励制度的一个关键概念。这是一个十分微妙却十分强大的设计。这个设计激励节点想方设法让其他节点延展它们自己的区块。因此如果网络中大部分节点遵循去延展最长支链的规则，那这样的设计将激励所有节点去遵循这个规则。这就是比特币的第一个奖励机制。

我们前面提到每产生 210 000 个区块（大约 4 年），区块奖励将被减半。在图 2.4 中，曲线的斜率将持续减半。这是一个等比数列，你可能知道数列的总和是有上限的。最终一共是 21 000 000 个比特币。

注意，这是新比特币被允许创造出来的唯一途径，没有任何其他新增币的机制。所以这是为什么比特币最终的数量是 2 100 万（至少目前的规则规定是这样）。按照现在奖励发放的速度，到了 2140 年比特币区块奖励就发完了。这是否意味着这个系统到了 2140 年就无法继续运行，并且因为不再有奖励诚实行为的激励而变得不安全呢？不是这样的。因为区块奖励只是比特币两种奖励机制之一。

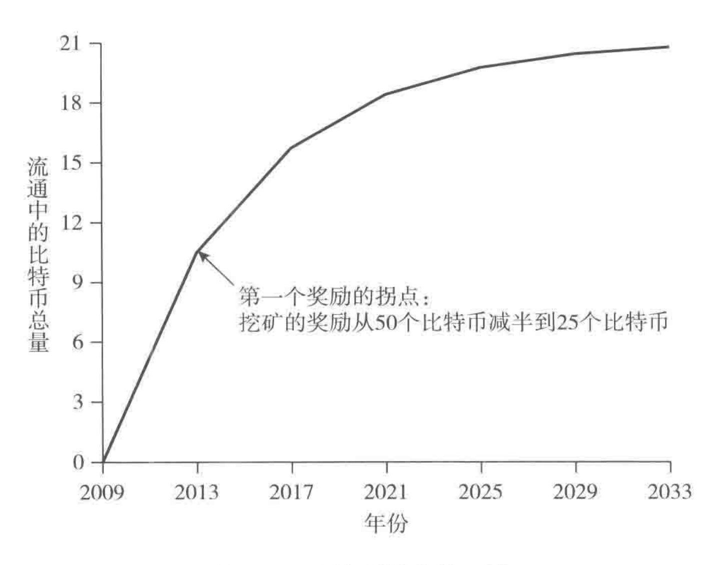


图 2.4 比特币的总供应量

注：区块奖励每 4 年减半一次，限制了比特币的供应上限为 2 100 万个。这是一个简化的模型，实际中的曲线会有轻微的差异，但都有 2 100 万的上限限制。

交易费

比特币的第二个奖励机制称为交易费。任何交易的制造者都可以选择让交易输出值比输入值小。第一个创建区块把交易放进区块链的人可以取得这个差额，作为交易费。如果你是一个节点，正在创建一个包含 200 笔交易的区块，那么这 200 笔交易的交易费将会被付到你放在区块内的那个地址。这些交易费现在是完全自愿的，但是我们可以预见，随着区块奖励逐渐发完，交易费会变得越来越重要，几乎是必需的，因为用户需要通过交易费来保障合理的服务质量。从某种程度上来说，这已经开始发生。但目前还不清楚这个系统会如何演变——这取决于还并不完善的博弈论的研究与发展。这也是比特币一个很有趣的研究领域。

我们在描述共识机制时还有一些问题没有解答。第一个主要问题是我们要你相信这样的假设：我们能随机选取一个节点。第二个是因奖励那些节点参与而产生的新问题：如果大家都想来分一杯羹成为一个节点来获得这些奖励的话，整个系统会变得不稳定。第三个是第二个问题的复杂版，就是可能会有攻击者

创建大量的女巫节点来尝试颠覆整个共识过程。

挖矿与工作量证明

事实证明，这些问题都是相互联系的，所有这些问题都有相同的解决办法——**工作量证明**（proof of work）。工作量证明的核心理念是，我们把随机选取节点改为根据节点占有某种资源的比例来选取节点，我们希望这种资源是没有人可以垄断的。比如说，如果这个资源是计算能力，那我们称之为工作量证明系统。或者这个资源可以是某种币的拥有量，我们称之为**权益证明**（proof of stake）。虽然比特币没有使用，但权益证明也是另一种合格的模式，并被其他加密货币所使用。我们将在第 8 章中更多地讨论权益证明和其他工作量证明的变种。

先回到工作量证明。我们试着更好地理解根据计算能力来选择节点到底是什么意思。换一个角度理解，我们是允许节点用它们的计算能力来互相竞争，导致的结果是计算能力的比例决定了节点被自动选中的概率。工作量证明还有一种理解方式，即我们把制造新身份的难度适度提高了。这就好像是对制造新身份，继而对女巫攻击收税一样。这听起来好像有点模糊，我们接下来看下比特币使用时工作量证明体系的细节，事情就变得清楚多了。

比特币是用哈希函数解谜来证明工作量的。任何一个提议并创建区块的节点想要制造下一块，这个节点必须要找到一个数，或者我们把它称为临时随机数（见第 1 章 1.1 节）。当你把这个临时随机数、前序块的哈希值还有要填进这个区块的交易列表连接起来，组成一整串字符，然后用哈希函数计算这一整串字符的输出值，这个输出值正好要落在一个相对于这个哈希函数所有可能的输出中很小的目标区间内。用公式来表示的话，就是临时随机数要满足下面的不等式：

$$H(\text{nonce} \parallel \text{prev_hash} \parallel \text{tx} \parallel \text{tx} \parallel \cdots \parallel \text{tx}) < \text{target}$$

就像我们前面看到的，通常一个区块会包含这个节点提议的一系列交易。而且，这个区块还会包含一个指向前序区块的哈希指针（我们这里说的哈希指

针是一个宽泛的概念。这个指针只是文本中的字符串，它并不需要告诉我们去哪里找到这个区块。我们可以通过在网上询问其他的节点找到区块。重要的是，这个哈希值既作为我们在网络上请求其他节点寻找区块的 ID，又能够让获取这个区块后验证它）。除此之外，我们现在还要求区块包含一个临时随机数。这个想法是为了适度提高发现符合要求的临时随机数的难度，即把包含临时随机数在内的整个区块的哈希值组合到一起，输出结果要是一种特定的形式。如果哈希函数符合我们在第 1 章中所描述的谜题友好特性，那唯一解出哈希谜题的办法就是去试足够多的临时随机数，直到成功为止。具体来说，如果这个目标区域是所有可能的输出的 1% ，那你大概就要试 100 次才能成功。事实上，这个目标区域远比输出范围的 1% 小得多得多，我们后面就能看到。①

用这种哈希函数解谜以及工作量证明的办法，我们可以完全舍弃采取那种随机选取节点的办法。这些节点在竞争哈希函数解谜的过程中一直都是互相独立的。有时一个节点鸿运当头，正好发现一个临时随机数可以满足要求。这个幸运的节点就可以提议创建下一个区块了。这就是比特币系统实现完全去中心化的方式，没有任何人能决定谁可以提交下一区块。

难于计算

哈希谜题有三个重要的特性。第一个特性是要有一定的难度。我们前面说适度的难度，其实你马上会看到难度实际上是随时间而改变的。在 2014 年年底，产生一个区块平均要做 10^{20} 次哈希运算。换言之，目标区域仅仅是整个输出范围的 $1/10^{20}$。这是超大的计算量——举例来说，超过了商业化笔记本电脑可能的计算范畴。因此，只有一些节点还在不厌其烦地竞争造块。这个不停尝试解哈希谜题的过程，就是我们听说的比特币挖矿，参与挖矿的节点被称为矿

① 这段话很拗口，也难懂。打个比方，假如你是个炮兵，那些前序区块哈希值加上所有要打包的交易就是炮弹，哈希函数就是大炮，临时随机数就好比瞄准器，你所想击中的目标，比如一个指挥所，它肯定在你能打到的范围内，但非常小，而且其实根本不知道在哪里。击中目标的唯一办法是狂轰滥炸，这就是比特币工作量的概念，炸的越多，击中的概率越高。如果这个指挥所的目标区域是轰击区域 1% 大的话，你大概平均要发出 100 枚炮弹才可能击中目标。——译者注

工。尽管技术上每个人都可以成为矿工，但由于挖矿的高成本导致了挖矿生态系统要消耗大量能源。

可参数化成本

第二个特性是，我们希望成本是可以通过参数来变化的，而不是一个固定值。在比特币的点对点网络里，是这样来达到这一特性的：每产生 2 016 个区块之后，所有的节点都会自动重新计算目标区域相对于整个输出范围的比例大小，使得后续的区块产生的时间间隔约为 10 分钟。两个区块之间的平均间隔是 10 分钟，2 016 个区块就需要两个星期。所以大约每两个星期，目标区域的大小会被重新计算一次。

我们想一下这意味着什么。如果你是个矿工，你花了一定的费用投资了一些硬件来做比特币挖矿。但是整个挖矿体系在不断增加，越来越多的矿工加入这个行业，或是他们部署了运算越来越快的硬件设备，那两个星期的时间段里，被找到的区块可能比预期的要多一点。然后，那些节点就会自动调整目标区域，你要找到一个块所要做的工作量就随之增加。所以如果你投了一笔固定资金在硬件上，你找到下一区块的速率实际上取决于其他矿工在做什么。有一个公式可以很好地描述这一点：任何一个矿工，比如爱丽丝，找到下一区块的概率，就相当于她控制的计算力占整个全球计算力的比例。这意味着，如果爱丽丝的挖矿设备的计算能力占全部计算能力的 0.1%，那大概每产生 1 000 个区块，她就可以找到一个区块。

这样重新调整的目的是什么？我们为什么想要维持 10 分钟间隔不变？原因很简单。如果区块产生的间隔太小，就会造成很多低效率，我们还会失去许多优化上的好处，比如在一个区块内放入大量的交易。10 分钟并没有神奇之处，如果把 10 分钟下调到 5 分钟大概也可以。关于其他加密货币的理想区块间隔应该是多少，已经有很多讨论。除去关于理想间隔的不同意见，大家都认为应该是个固定的值。它不允许被无限降低。这就是为什么我们有自动重新计算目标区域的特征。

这个成本函数和工作量证明的设定方式，让我们重新审视比特币的安全假

设。现在我们终于可以丢弃之前让你盲目相信的假设。不必再去说那些连身份都没有的节点大多数是诚实的了，诚实具体代表什么也并不清楚，我们现在可以清楚地表述，只要以计算能力为权重的大多数矿工，遵循比特币协议，或者说是诚实的，那么比特币中的大量攻击就都没有可能发生。因为如果以计算能力为权重的大多数矿工是诚实的，提议下一个区块的竞争会自动保证在任意时间点，下一个区块至少有50%的概率是由一个诚实节点提议的。

矿工行为的两种行为模式

在分布式系统和计算安全研究领域，假设一定比例的节点是诚实的，来展示在其他节点表现随意的情况下，系统如何按照预期运行，是很常见的方法。这是我们采用的基础方法，除了以计算能力为权重计算大多数之外。最初的比特币白皮书也包含了这样的分析。

但博弈论领域给出了一种完全不同的，更复杂且实际的方法来决定系统如何运行。这个观点不区分节点诚实或恶意，而是假设每个节点都按自己的意愿行动。每个节点考虑其他节点的潜在可能策略之后，采用一种（随机的）策略最大化自己的回报。如果协议和激励机制设计得当，大多数节点在大多数时候会遵循这个规则。“诚实”的行为只是许多策略中的一种，我们在道德上并不依赖于此。

博弈论的观点认为，最大的问题是矿工默认的行为是否是一种“纳什均衡”（Nash equilibrium），即这是否代表了一种稳定的状态，在这种状态下没有节点可以通过表现不诚实而获得更高的回报。针对这个问题现在各界仍有争议，并且是一个活跃的研究领域。

解哈希谜题是概率性的，因为没有人可以预测到哪个临时随机数会解出谜题。唯一的方法是一个一个去试临时随机数，并希望能够成功。在数学上，这被称为伯努利试验（Bernoulli trial）。伯努利试验是一种有两种可能结果的试验，在连续试验下，每种结果发生的概率是固定的。在这里，两种结果是哈希值是

否落在目标区域内，假设哈希函数像随机函数一样，那些结果的概率都是固定的。典型地，节点多次尝试临时随机数的伯努利试验是一个离散概率过程，它可以用一个叫作泊松过程（Poisson process）的连续概率过程近似表示，在泊松过程中，事件以固定的速率独立出现。最后的结果是，发现下一个区块所需要时间的概率密度函数，见图 2.5。

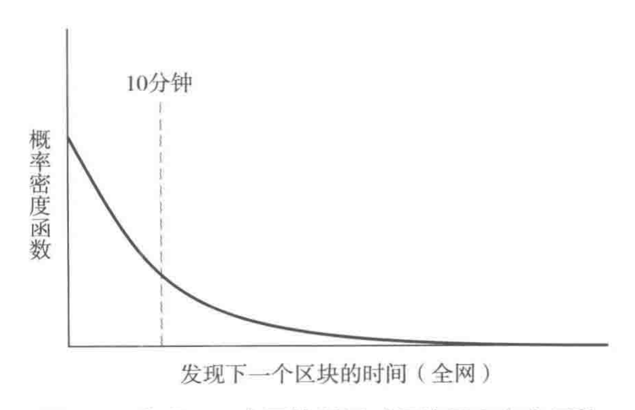


图 2.5　发现下一个区块所需时间的概率密度函数

这被称为指数分布。假设一个区块现在被发行，下一个区块有一定的小概率很快被发现，比如几秒钟或几分钟。也有一定的小概率花了较长时间才发现下一个区块，比如一小时。但总体来说，网络会自动调整难度使得区块间隔时间的长期均值维持在 10 分钟。注意图 2.5 表示的是整个网络内区块被创造出来的频率，而不是哪个矿工事实上发现了这个区块。

如果你是一名矿工，你大概想知道要多长时间才能找到下一区块？这个概率密度函数会是什么样？它的形状会相同，但 x 轴的坐标不一样。可以用一个漂亮的公式表示：

对于某个特定的矿工：

发现下一区块的平均时间 = 10 分钟/占全部计算能力的比例

如果你有全网络 0.1% 的计算能力，这个公式告诉我们，你每 10 000 分钟能找到一个区块，大约一个星期。不仅是你发现区块的时间间隔非常长，时间间隔的波动也会非常大。因此产生的一些重要结论我们在第 5 章会讲到。

易于证实

现在回到工作量证明函数第三个重要的特性，就是证实一个节点正确地计算了工作量证明很容易。即使一个节点要尝试 10^{20} 次来找到使区块哈希值落在目标范围内的临时随机数，并且临时随机数必须是作为区块的一部分被公布出来。这样任何其他节点很容易检查区块的内容，计算它的哈希值，证实它的输出在目标区域内。这是个相当重要的特征，因为这样使得我们摆脱了中心化管理。我们不需要一个中央权威机构来证明矿工正确地完成了工作。任何节点或者矿工，都可以迅速地证实其他矿工找到的区块符合工作量证明的规定。

2.5 总结

挖矿成本

我们现在来看一下挖矿经济学。前文提到过，作为矿工挖矿是十分昂贵的。按现在的难度，找到下一个单独的区块需要计算 10^{20} 个哈希值，区块奖励约是 25 个比特币，按照现在的比特币汇率，是不小的一笔钱。这些数据可以让我们简单地计算出挖矿是否赚钱。我们可以用这个简单的逻辑来做这个决定：

如果：
挖矿奖励 > 挖矿成本
那么：
矿工赚钱
条件是：
挖矿奖励 = 区块奖励 + 交易费
挖矿成本 = 硬件成本 + 运营成本（电费、空调费等）

基本上，矿工的挖矿奖励就是区块奖励和交易费。矿工自己与总的支出相比较，包括硬件和电费成本。

但这个简单的公式也有几个复杂的地方：第一，硬件投资是固定的，但电费是个变量，随时间变化。第二个复杂之处是，矿工得到的奖励取决于他们发现区块的速度，这不仅取决于他们硬件的能力，还取决于他们的计算能力占全球计算能力的比例。第三是挖矿产生的成本通常是用美元和其他传统货币表示的，但他们得到的奖励是比特币。所以这个方程在任何时候都有一个隐藏的因素，就是比特币的汇率。第四，到目前为止，我们都假设矿工会诚实地遵守协议。但矿工有可能选择用一些其他的挖矿策略，而不总是试图延展最长的有效分支。所以这个方程没有囊括所有矿工可以用到的不同策略的细微差别。事实上，要想分析挖矿是否有意义，是一个博弈论问题，没有那么容易找到答案。

到此为止，我们已经较好地理解了比特币如何实现去中心化。我们现在总结一些关键点，放在一起以便更好地理解。

我们首先从身份开始。根据我们知道的，比特币协议不需要真实世界的身份就可以参与。任何用户任何时刻都可以制造一对匿名的钥匙。如果爱丽丝想付给鲍勃比特币，比特币协议里没有详细说明爱丽丝如何得知鲍勃的地址。在这些匿名的钥匙对用作身份的情况下，交易其实是向整个点对点网络广播的信息，把比特币从一个地址转到另一个地址。比特币只是交易输出，我们在下一章节会深入讨论这个问题。

不存在所谓“1 比特币”这样的东西

比特币没有固定面额，不像美元。具体来说，没有“1 比特币”这样的特别名称。比特币只不过是交易输出，在现在的规则里，它们可以是精确到小数点后 8 位的任意值。可能的最小价值是 0.000 000 01 BTC（比特币），我们称之为 1 个中本聪（比特币的发明人）。

比特币点对点网络的目标，是把所有新的交易与新的区块传播到所有比特币节点。但这个网络很不完美，只能尽其所能来传递信息。这个系统的安全性不是来自点对点网络的完美，而是来自我们本章中重点讨论的区块链和共识协议。

当我们说一个交易被放进了区块链，我们真实的意思是这笔交易已被确认了许多次。对于多少次确认足以让我们确信交易已包含在内，并没有一个固定的数字，但 6 次是个常用的数目。一笔交易收到的确认越多，你就越可以确信这笔交易被放进了区块链。经常会有一些孤块，或者没有进入共识链的区块。有很多原因可以导致一个区块变为孤块。这个区块可能包含一个不正当交易，或者试图双重支付。也有可能是网络延迟，这里指的是，两个矿工可能相隔几秒找到了新的区块，这两块几乎同时被广播到网上，那其中一块肯定会被丢弃。

最后我们看了哈希谜题与挖矿。矿工是决定参与创造新区块竞争的特殊类型节点。如果其他矿工继续在他们的区块上搭建的话，对于他们努力的回报是新造的比特币（新区块奖励）和已经存在的比特币（交易费）。很微妙也很重要的一点是：假设爱丽丝比鲍勃的计算能力要强大 100 倍，但这并不意味着爱丽丝一定能够赢得找到下一区块的竞赛，而是爱丽丝和鲍勃发现新区块的概率比率是 100∶1。长期下来，鲍勃找到的区块数量是爱丽丝的 1/100。

我们预计矿工们会处在经济平衡点附近，意味着他们得到的奖励大致等于他们在硬件与电费上的花费。理由是如果一个矿工持续亏钱，他会停止挖矿。反之，如果硬件和电费固定的情况下，挖矿利润很高，那更多的挖矿设备会加入网络。计算能力的增加会导致难度提高，每个矿工预期的回报便会降低。

比特币深度使用了分布式共识的概念。在传统货币系统中，共识的作用是有限的。具体来说，有一个共识过程来决定货币的汇率。这在比特币里当然也是对的——我们需要围绕比特币价值的共识。但在比特币里，我们还需要对账本情况的共识，这是由区块链来完成的。换句话说，甚至你拥有多少比特币的算法都是依赖共识的。当我们说爱丽丝拥有一定数量的比特币，我们真实的意

思是说在比特币点对点网络，在区块链中记录的所有爱丽丝地址上拥有的比特币数量总额。这是比特币系统的一个终极真相：拥有比特币就是其他节点对给定的一方拥有这些比特币的共识。

最后，我们必须对整个系统的规则达成共识，系统规则有时不得不改变。比特币规则改变有两种：对应为软分叉与硬分叉。我们把关于它们区别的详细讨论放到第 3 章和第 7 章中。

启动加密货币

另一个微妙的概念是“自举过程”（bootstrapping）。比特币系统里三个不同的想法——区块链的安全性、挖矿生态系统的健康程度，以及货币的价值在相互作用。我们显然希望区块链安全，这样比特币才能成为一种可行的货币。想要区块链安全，就要保证黑客不能倾覆共识过程。这反过来意味着，一个黑客不能够制造一大堆挖矿节点来占据 50% 以上的新区块生成。

但如何实现这一点呢？前提条件是有一个健康的挖矿生态系统，其中大部分节点是诚实的、遵守协议的。但健康的挖矿生态系统的前提条件又是什么呢——我们什么时候可以保证大多数矿工会把大多数计算能力运用到解哈希谜题的竞争中呢？好吧，只有在比特币价位高时他们才会这么做，因为他们收到的奖励是比特币而他们的花费都是美元。所以币的价值越高，矿工就越有动力这么做。

那如何保障币的价值又高又稳定呢？只有用户普遍相信区块链的安全性才会发生。如果他们认为网络随时会被攻击者颠覆，那比特币作为货币将一文不值。所以你可以看到区块链的安全性、挖矿生态系统的健康程度和货币的价值这三者之间相互依赖、相互作用的关系。

因为这三者之间的循环依赖关系，其中一个的存在可以用另一个的存在推测出来。在比特币初创之时，这三者都不存在。除了中本聪自己，没有人在运行挖矿软件。比特币作为货币没有什么价值。事实上，因为没有很多人挖矿，区块链也很不安全，任何人都可以轻易颠覆这个过程。

这三者在比特币的世界如何从无到有并没有一个简单的解释。媒体的关注

是其中一个因素——听到比特币的人越多，感兴趣挖矿的人就越多。挖矿的人越多，人们就会对区块链的安全越有信心，因为更多挖矿活动在进行，以此类推。附带提下，每种其他虚拟货币想要成功也需要想办法通过自举过程解决这个问题。

51%攻击

我们考虑一下如果共识失败，并且存在一个在比特币网络里实际掌握了绝大部分挖矿计算能力的51%攻击者，会发生什么情况。我们考虑多种可能的攻击，分析哪些可能被这样的攻击者实际使用。

首先，攻击者可以从现存的地址里偷币吗？你可能猜到了，不行，因为除非你能推翻加密方法，否则从现存地址偷币是不可能的。它不足以颠覆共识过程。这样说还不是很清楚。我们不妨假设，51%攻击者制造了一个不正当的区块，里面有一笔不正当交易把币从不受其控制的地址转移到自己的地址。攻击者可以假装这是一笔正当的交易，继续在这个区块上建造，甚至可以把它变成一个最长的支链。但其他诚实节点不会简单地接受这个存在不正当交易的区块，它们还是会在网络中找到之前最后一个正当的区块，基于此继续挖矿。所以将会发生的是，链上出现了我们称之为分叉的情况。

现在想象一下这个攻击者想把这些非法的币花掉，付给某个商家鲍勃用来买他的商品或者服务。鲍勃可以假定运行着自己的比特币节点，而且是一个诚实节点。那鲍勃的节点会因为含有不正当的交易而拒绝这个非法的分支。因为那里面的数字签名不吻合。所以鲍勃的节点会忽略这个最长的支链，因为这是一个非法的支链。而因此，这不足以颠覆共识。你需要推翻加密方法偷取比特币。所以我们认为，这个攻击对51%攻击者来说是不可能的。

我们应该注意到这是一个想象的实验。如果实际中真的有51%攻击的迹象，可能会发生的是开发者会注意到并采取应对措施。他们会升级比特币软件，我们可以期待系统规则（包括点对点网络）可能会做出改变，使得这样的攻击难以成功。但我们无法准确预测。所以，我们是在一个简化的模型上讨论51%攻击，但除此之外系统规则并没有改变和扭曲。

我们考虑另一种攻击。51% 攻击者可以压制其他交易吗？比如攻击者特别讨厌某个用户卡罗尔。他知道卡罗尔一些地址，想使属于这些地址的币都无法使用。这可能吗？由于攻击者控制了区块链的共识过程，他可以轻易地拒绝创造包含来自卡罗尔地址的交易的新区块，他还可以进一步拒绝在含有类似交易的区块上延展。但他不能阻止这个交易被广播到整个点对点网络，因为网络不依赖于区块链或者共识，我们假设攻击者还没完全掌控网络。他不能阻止这个交易被发送到绝大部分节点上，所以即使他成功了，大家也都知道发生了攻击。

攻击者可以改变区块奖励吗？比如说攻击者开始假装把区块奖励由 25 个币改成 100 个币？这是对系统规则的改动，因为他没有控制所有诚实节点上运行着的比特币软件备份，所以同样不可能。这和为什么攻击者无法装入一笔非法交易的道理是一样的。其他节点不会轻易认可区块奖励提高，所以他也无法使用这些区块。

最后，这个攻击者会摧毁大家对比特币的信心吗？好吧，让我们想象一下会发生什么。如果有很多双重支付尝试，诸如节点不延展最长的有效分支，以及发生其他攻击，那么人们有可能会觉得比特币不再是一个他们可以信赖的去中心化账簿。人们会对货币失去信心，我们可以预料到比特币汇率会重挫。实际上，如果人们知道有一方控制了 51% 的哈希算力（hash power），即使这个人没有发动任何攻击，大家也可能会对比特币失去信心。所以，这不仅仅是可能，事实上任何形式的 51% 攻击都会摧毁大家对货币的信心。这其实是 51% 攻击可以实现的最主要的实际威胁。考虑到在攻击比特币，实现 51% 多数的过程中，财政角度的巨大花费，我们讨论的这些攻击都会变得不切实际。

我们希望至此你对比特币的去中心化管理有了一个完整的了解。你也应该理解了比特币里的身份如何工作、交易是如何被传播和验证的、比特币里点对点网络的作用、如何用区块链达成共识、函数难题与挖矿是怎么回事。这些概念为理解比特币的更多微妙细节和细微差别提供了坚实的理论基础，是一个良好的出发点。这些我们在后续章节中会进一步看到。

延伸阅读

比特币白皮书：

Nakamoto, Satoshi. "Bitcoin: A Peer-to-Peer Electronic Cash System." 2008.

下载地址：https://bitcoin.org/bitcoin.pdf.

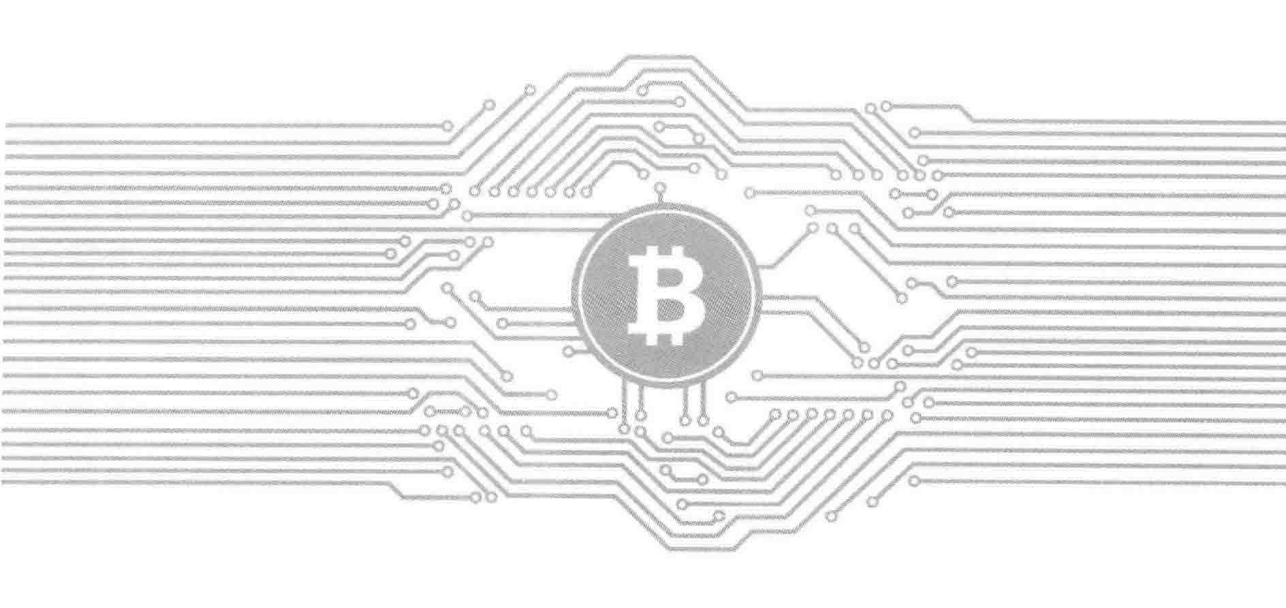
最初的基于工作量证明的介绍：

Back, Adam. "Hashcash—A Denial of Service Counter-measure." 2002.

下载地址：http://www.hashcash.org/papers/hashcash.pdf.

Paxos 共识算法介绍：

Lamport, Leslie. "Paxos Made Simple." *ACM Sigact News* 32 (4), 2001: 18-25.



第3章

比特币的运行机制

BITCOIN
AND
CRYPTOCURRENCY
TECHNOLOGIES
A Comprehensive Introduction

在这一章我们重点剖析一下比特币的运行机制。前两章里，我们是在相对泛泛的层面讨论了比特币，在这一章中我们将深入细节，真正近距离地了解比特币所使用的数据结构、实际脚本以及语言，这种较为精准的介绍会为本书后面的章节建立场景。这一章会有大量细节性的信息，极具挑战。本章可以帮助我们真正懂得比特币的实质。

如第2章所述，比特币的共识机制设定了一个只允许往里写入的账簿，而且一旦数据被写入，它将永远被储存在那里。去中心化（或者分布式）协议确保了账簿中存储数据的共识，而矿工会执行协议并确认交易，这些机制可以确保每一笔交易都是真实发生的，而且账簿中的每一个比特币都没有被使用过。这样，这个账簿从功能上就形成了一种货币系统。同时，我们也假设，可以使用货币奖励矿工，使矿工有积极地完成记账操作的动力。在本章中，我们将详细介绍如何建立货币系统、如何奖励矿工，从而保证整个流程有序运行。

3.1 比特币的交易

让我们先一起看一下比特币的交易，比特币交易的过程其实就是不停地创造区块的过程，为了理解上的方便，我们先看一个简单模式的账簿，在这个账簿里，每一笔交易依次被添加到账簿里。

那我们如何使用这个账簿来创造一种货币呢？首先你可能想到（也是许多

人误认作比特币使用的方式)：建立一个以账户为核心的系统，可以创造新的币并且放入某人的账号，然后就可以转给其他人了。一笔交易的信息就像这样："把爱丽丝账户里17个币转给鲍勃"，然后由爱丽丝签名。我们从图3.1可以看到，爱丽丝在第一笔交易里收到25个币，然后转了17个币给鲍勃，她的账户里应该还有8个币。

系统制造了25个币给爱丽丝，由矿工确认
爱丽丝转了17个币给鲍勃，由爱丽丝签名
鲍勃转了8个币给卡罗尔，由鲍勃签名
卡罗尔转了5个币给爱丽丝，由卡罗尔签名
爱丽丝转了15个币给戴维，由爱丽丝签名

图3.1　基于账户的账簿

这么做的不便之处在于，任何人如果想要确认一笔交易是否真实，就必须跟踪每一个账户的余额。让我们再看一下图3.1，当爱丽丝想要转给戴维15个币的时候她是否真的有15个币呢？为了搞清楚这个问题，我们必须回过头去看和爱丽丝有关的所有交易，并加总来确认当时的余额。当然，我们可以有一些更有效的办法，比如另外增加一个数据字段，用来更新每次交易后的账户余额，但这也增加了记账的工作量。

所以，比特币并没有用这种记账方式，而是用了我们在第1章1.5节里提到的"财奴币"相类似的方法来记录交易。

这种方式就像财奴币里的付币，每个交易中都有一个输入值和输出值。输入值可以看成是将被消费掉的币（这些币是前一个交易创造出来的)，把输出看成是在本次交易中创造出来的币。铸造新币时，只会创造新币，而不会消费旧币（就像财奴币里的造币，只有输出，没有输入)。每笔交易都有一个独一无二的ID。每笔交易中可能有多个输出，输出的索引从0开始，所以我们称第一个输出为"输出0"。

我们现在来看图3.2。交易1是铸造新币的交易，因此没有输入，也没有签名；交易1的输出是向爱丽丝转移25个币。现在，爱丽丝想把一些币转给鲍勃，她就创造了一条新的交易，这就是图3.2中的交易2。在交易里，她必须明

确指出要转出的币的来源（引用之前的某笔交易）。爱丽丝指出本次交易的币来自交易 1 中的输出 0（也是交易 1 中的唯一输出），即向爱丽丝转移 25 个币。交易中，爱丽丝还要明确收款人——也就是输出的地址，在这个例子里，有两个输出，一个是转 17 个币给鲍勃，另一个是转 8 个币给爱丽丝自己。当然，整个交易由爱丽丝签名，这样，大家就知道这笔交易爱丽丝是确实授权了的。

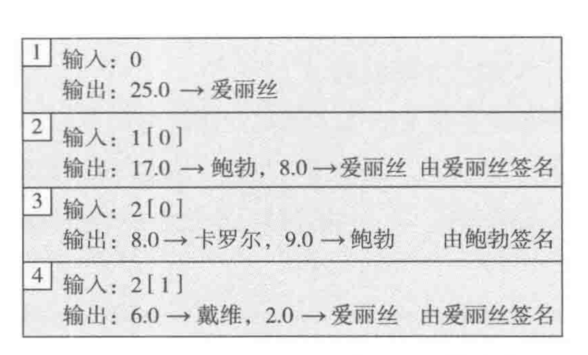


图 3.2　与比特币类似的基于交易的账本

地址转换。在这个例子里，为什么爱丽丝要把币转给自己呢？事实上比特币就像财奴币中描述的币一样，一个交易中输出的币，要么在另一个交易中被完全消费掉，要么就一个都不被消费，不存在只消费部分的情况。爱丽丝只需付给鲍勃 17 个币，但爱丽丝在上一交易中实际获得了 25 个币，为了把这些币全部消费掉，她必须再转给自己 8 个币。这 8 个币可以转到另外一个地址（不同于交易 1 中获得 25 个币的地址），但前提是该地址为爱丽丝所有，这就叫地址转换。

有效验证。当一个新的交易被加入总账，它的有效性是否容易被验证？在这个例子里，我们要核查一下爱丽丝引用的交易输出，确认她确实有 25 个币没有被花费掉。因为我们使用了哈希指针，所以核查很快。为了确认这 25 个币没有被花掉，我们只需从爱丽丝所引用的交易开始，一直核查到账本上最新记录的交易为止即可——而不需要从账本建立之初的交易开始核查。而且，这种方法也不需要增加额外的数据结构（当然，我们将会看到，加入新的数据结构将进一步提高速度）。

资金合并。和财奴币一样，比特币交易可能有许多输入与输出，资金分隔与合并也很容易。假如鲍勃在两笔不同的交易中分别收到 17 个币和 2 个币，现

在他想把这两笔钱合并起来花掉，这很容易，他只需发起一个交易，交易里有两个输入和一个输出，输出的地址是他自己的地址，这样，鲍勃就把两个交易合二为一了。

共同支付。同样地，共同支付也很容易做到。如果卡罗尔和鲍勃想要共同支付给戴维，他们可以发起一个交易，交易里也有两个输入和一个输出，唯一不同在于，两个输入所引用的“上一笔交易”的输出地址不同，因此，这笔交易需要两个签名：卡罗尔的和鲍勃的。

交易语法。比特币交易涉及的概念就是上面这些。我们再来看看比特币交易在底层是如何实现的。实际上，比特币在网络上传输的数据结构都是一串字符，图 3.3 显示了一个真实的程序，经过编译就会变成供机器执行的二进制代码了。

元数据 / 输入 / 输出

```
{
  "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
  "ver":1,
  "vin_sz":2,
  "vout_sz":1,
  "lock_time":0,
  "size":404,
  "in":[
   {
      "prev_out":{
      "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
      "n":0
     },
       "scriptSig":"30440..."
   },
   {
      "prev_out":{
      "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
      "n":0
     },
     "scriptSig":"3f3a4..."
    }
   ],
   "out":[
    {
      "value":"10.12287097",
      "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e
            OP_EQUALVERIFY OP_CHECKSIG"
    }
   ]
}
```

图 3.3　一个真实的比特币交易程序段

从图3.3可以看到，一个比特币交易分成三部分：元数据、一系列的输入和一系列的输出。

- 元数据。这里存放一些内部处理的信息：包含这笔交易的规模、输入的数量、输出的数量，还有此笔交易的哈希值，也就是这个交易独一无二的ID。我们可以用哈希指针指向这个ID。最后还有一个“锁定时间”(lock_time)，我们后面会谈到。
- 输入。所有输入排成一个序列，每个输入的格式都是一样的。输入需要明确说明之前一笔交易的某个输出，因此它包括之前那笔交易的哈希值，使其成为指向那个特定交易的哈希指针。这个输入部分同时包括之前交易输出的索引和一个签名：我们必须有签名来证明我们有资格去支配这笔比特币。
- 输出。所有输出也排成一个序列。每个输出的内容分成两部分。所有输出的金额之和必须小于或等于输入的金额之和。当输出的总金额小于输入总金额时，输出的总金额与输入的总金额的差额部分，就作为交易费支付给为这笔交易记账的矿工。

一长串怪怪的（funny）字符看上去像是接收地址。实际上，每个输出都要和一个特定的公钥（地址）对应，所以这一长串字符里面确实有一部分看上去是公钥的哈希值，但里面还有一部分看上去像指令集合的东西，它其实是一个比特币的脚本，下文展开介绍。

3.2 比特币的脚本

每个交易输出不仅确定了一个公钥，其实同时指定了一个脚本。那脚本是什么？为什么我们要用一个脚本？在这一节我们要学习比特币的工作控制语言，也叫脚本。之后，我们就会懂得为什么要用一个脚本，而不是简单地分配一个公钥。

最常见的比特币交易，就是通过某人的签名去取得他在前一笔交易中获得的资金。这种情况下，我们希望交易的输出包含这样的信息："凭借地址 X 的所有者的签名，才可以获得这笔资金。"我们知道地址其实就是一个公钥的哈希值，所以仅仅说地址 X 并没有告诉我们公钥在哪里，也没有给我们一个检查签名的方法。所以，交易输出必须这样描述："凭借哈希值为 X 的公钥，以及这个公钥所有者的签名，才可以获得这笔资金。"这实际上就是最常见的比特币脚本，如图 3.4 所示。

```
OP_DUP
OP_HASH160
69e02e18...
OP_EQUALVERIFY
OP_CHECKSIG
```

图 3.4　P2PH 脚本范例

注：一个常见的比特币输出脚本范例。

那么谁执行这个脚本？这一系列指令是如何完成的呢？秘密在于，交易的输入包括了脚本（而不是签名）。为了确认一笔交易正确地获取了上一笔交易所输出的资金，我们把交易的输入脚本和上一笔交易的输出脚本串联起来，这个串联脚本必须被成功地执行后才可以获取资金。这两个脚本，一个是输出脚本（scriptPubKey），另一个是输入脚本（scriptSig）。输出脚本只是指定了一个公钥（或是公钥哈希值的地址），输入脚本指定了一个对应公钥的签名。图 3.5 就是两个脚本结合的案例。

比特币脚本语言

这个脚本语言是为比特币开发的。在比特币里只叫作"脚本"。它和另一种 Forth 语言有很多相似的地方，Forth 是一种简单的堆栈式编程语言（stack-based programming language），但你并不需要先学习 Forth 语言才会使用比特币的脚本语言。比特币的脚本语言设计原则就是简明扼要，并内生地支持加密操作。比如，脚本里面有目的性的指令用来计算哈希值和检验签名。

这种脚本语言是堆栈式的，意味着每个指令只被执行一次，是线性的，无法循环执行。所以指令的数目给了我们一个执行时间与内存使用的上限。这个

语言不是图灵完备的，意味着不能随意运行强大函数功能。[①] 但这是有意设计的，因为矿工需要去执行这些网络上任意交易提交者所递交的脚本，设计者并不希望让他们提交可能无限循环的脚本。

```
<sig>
<pubKey>
--------------
OP_DUP
OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG
```

图 3.5 结合输入脚本和输出脚本范例

注：为了确认当前交易是否正确地获取了前一笔交易输出的资金，我们把两个脚本链接起来，把上一笔交易的输出脚本（图中虚线下方）添加到当前交易的输入脚本（虚线上方）之后，形成一个新的脚本。请注意 < pubKeyHash? >里面有一个“?”，用作标识——我们后面会来确认它是否与当前交易提供的公钥的哈希值一致。

执行比特币脚本只能产生两个结果：要么被成功执行，这种情况下，交易有效；要么脚本执行出现错误，这种情况下，整个交易无效，拒绝记入区块链。

这个脚本语言十分简单。只有 256 个指令，每个只用一个字节。256 个指令中，有 15 个目前不可用，有 75 个被保留还没有具体定义（以后或许可以被用来扩展），剩下的才是可用的。

许多在其他语言里常见的基本指令这里面都有。例如，基本的算数、逻辑语句（如 If-then）、抛出错误、过早返回等。而且，还有密码指令，比如哈希函数语句、签名验证语句，还有一个重要的特殊指令是“CHECKMULTI-SIG”——可以查证多个签名。表 3.1 列举了一些比特币工作控制语言里的常用语句。

CHECKMULTISIG 指令要求指定 n 个公钥和一个参数 t（作为一个临界值）。这个指令正确执行的条件是：在 n 个公钥中，至少可以选出 t 个现时有效的签

---

① 图灵是第二次世界大战时英国数学家，密码学家。他破译了纳粹的密码机“谜”，为盟军取得第二次世界大战胜利做出重大贡献，美国好莱坞以此题材拍了一部电影《模仿游戏》。图灵完备的意思是语言有能力随意地执行强大的函数。——译者注

名。我们在本章3.3节会示范这个指令的用法，但现在我们需要认识到这个原生指令是非常强大的，它以一种极其精练的方式协助我们查验交易中的多方签名。

不过，目前比特币多方签名功能实现过程中有一个缺陷，CHECKMULTISIG指令在执行的时候会返回一个没用的值，而且系统还必须要安排一个堆栈中的变量去储存它，然后再忽略掉。由于修复这个缺陷成本很高，两害相权取其轻，这个缺陷就一直没被修复，我们在第3章3.6节会再做讨论。但目前，这个程序缺陷也算是比特币的一个特性。

表3.1　一些比特币脚本工作语言中的指令及其功能

指令名称	功能
OP_DUP	复制堆栈顶端数据
OP_HASH160	计算哈希函数两次：第一次用SHA-256，第二次用RIPEMD-160
OP_EQUALVERIFY	如果输入是相同的，返回真 如果输入是不同的，返回假，整个交易作废
OP_CHECKSIG	检查输入的签名是否有效
OP_CHECKMULTISIG	检查在交易中t个公钥（地址）对应的t个签名是否有效

执行一个脚本

在堆栈语言里执行一个脚本，我们只需要一个堆栈来垒积数据，不需要分配任何内存与变量。因此，堆栈语言中计算相当容易。总共有两类指令：数据指令和工作码指令。数据指令的作用是把数据推到堆栈的最上面；工作码指令则通常是用堆栈顶部的数据作为输入值，用来计算一个函数。

我们现在来一起看一下，图3.5这段脚本是怎么执行的。图3.6给我们展示了每一条指令执行后的堆栈状态。脚本中的前两条指令属于数据指令，分别是输入脚本（包含在交易的输入项）中的签名和用来验证签名的公钥。我们前面

提到过，一看到数据指令，系统就把它堆到堆栈最上面。后面几个指令是输出脚本（包含在上一交易的输出项中）里的指令。

首先，我们复制指令 OP_DUP，这一步仅仅是将堆栈最上层的公钥复制，并置于堆栈最上层；下一个指令是 OP_HASH160，该指令取得堆栈最上层的数据，并计算其哈希值，然后将结果再堆到堆栈最上层。当指令执行完成后，我们将堆栈最上层的公钥替换成了公钥的哈希值。

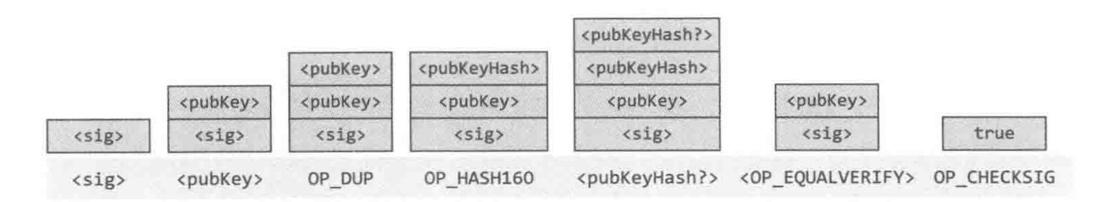


图 3.6　比特币脚本的执行堆栈状态图

注：图中底部列出了相对应的指令：尖括号里的是数据指令，以 OP 开头的是工作码指令，指令上方对应的是指令执行之后的堆栈状态。

接下来，我们还要在堆栈顶层再推送一些数据：此笔交易发送者指定的公钥的哈希值，以及对应的私钥，这样才可完成签名，取得资金。此时，堆栈顶部有两个数值，一个是发送者指定的公钥的哈希值，另一个是接收者想要取得资金时提交的公钥的哈希值。

这个时候，我们就要执行 EQUALVERIFY 命令了，这个命令是用来检查堆栈顶部两个数值是否相等的。如果不相等，就会抛出一个失败信号，并且停止执行脚本。不过现在我们假设其相等，也就代表着接收者使用的是正确的公钥。这条指令会移除堆栈顶部的两条数据，这时，堆栈还剩下两个数据：公钥以及签名。

我们已经证实接收者使用的公钥确实就是交易里指定的公钥，但现在我们必须证实这个签名是真的。这时，使用 OP_CHECKSIG 指令即可。这里我们可以看出比特币的脚本语言虽然简单，但很强大。它只用“OP_CHECKSIG”就能实现一个很复杂的事情：移除堆栈里两个数值，然后用公钥来证实整个交易的签名是真的。

但这里的签名究竟是对什么的签名？签名函数的输入是什么？实际上，在比特币中，我们只可以对一个事情进行签名——就是整个交易。所以，CHECK-SIG 指令从堆栈中取出两个数据（公钥以及签名），并验证签名对于整个交易（使用对应公钥发起的交易）来说是有效的。现在我们完成了所有的指令，堆栈里面什么也不剩。假设没有碰到任何差错的话，这个脚本的输出就是一个“真”表示这个交易是正当有效的。

实际情况

理论上来讲，通过脚本，我们可以随意地为比特币支付设定条件。当然，从 2015 年的情况看，这些特性也并不太常用到。如果我们回顾比特币历史中曾经实际用到的脚本，绝大多数的比特币使用的脚本都非常基础，像前文的例子一样：指定一个公钥，然后通过验证签名来使用这个币。

当然，实际中也会使用一些其他指令，比如 MULTISIG，还有一种支付给脚本的哈希值（Pay-to-script-hash，简称 P2SH，我们很快会谈到）等，但除此之外，平时常用的指令真不多，因为每个节点都有一份标准脚本的白名单，它们会拒绝接受不在名单上的脚本。这倒不是说无法运行其他脚本，只是使用起来比较麻烦。事实上这样的安排也很巧妙，我们会在谈论比特币点对点网络的时候再进行描述。

销毁证明

销毁证明（proof of burn）脚本，用于销毁比特币（即防止资金被赎回）。如果交易代码的运行结果是将比特币转到“销毁证明”脚本，那么这笔比特币将被销毁。实际应用中主要是用来引导客户使用其他数字货币系统，即将比特币销毁，以便获得另一个数字货币系统发行的新币。我们会在第 10 章展开叙述。销毁证明脚本使用起来非常简便：使用 OP_ RETURN 脚本来抛出错误；不论之前指令的运行结果是什么，OP_ RETURN 指令总会被执行，并相应抛出一个错误，脚本返回一个“错误”（false）值。

由于 OP_ RETURN 以抛出错误的形式结束脚本，其后的所有指令都不会执

行。利用这个特性，我们可以往脚本中植入任意信息，这些信息也将被存储在区块链中。假如你想通过署名或者盖时间戳的方式来证明你在某个时候知道某件事情，就可以发起一笔极小额的比特币交易，在脚本中加入上述信息，并使用销毁证明脚本将币销毁，这样就可以将信息永久地存储在区块链上。

支付给脚本的哈希值

如前文所述，比特币的工作机制要求币的发送者必须在交易时明确指定脚本。这种机制有时候不太适用：假如你在网店看中了一件商品并打算下单，你会问卖家“请把付款地址告诉我，我可以付款了”，但如果卖家使用了多重签名地址（MULTISIG），那他会说“嘿，我们用了多重签名地址，你需要支付给一个脚本地址，而不是一个简单的地址”，但你会说“我不知道怎么弄，这太复杂了，我只会支付给简单的地址”。

比特币用了一种很聪明的办法来解决这个问题，不仅可以实现多重签名地址支付，而且还可以实现复杂的资金监管规则。比特币使用的办法是：收款方告诉付款方“请把比特币支付给某个脚本地址，脚本的哈希值是××，在取款的时候，我会提供上述哈希值对应的脚本，同时，提供数据通过脚本的验证”，而不是“请把比特币支付给某个公钥，公钥的哈希值是××”。付款方通过P2SH即可实现上述交易。

需要说明的是，P2SH脚本只是对堆栈最顶层的数据进行哈希运算，核验运算结果是否与给定的哈希值一致，核验通过后，再执行一步特殊的核验：将堆栈最顶层的数据重新解读为一系列指令，然后将其作为脚本运行一次，此时，堆栈中的其他数据作为脚本的输入值。

要做到P2SH还是有点复杂的，因为P2SH不是比特币的原始设计，是后来加上去的。它解决了两个重要的问题：让付款方的支付工作简单化，收款方只需告诉付款方一个哈希值即可。在我们上面的例子中，你不再需要去关心商家到底用哪种地址，是否用了多重签名，因为这只是商家在支取这笔款项时需要考虑的事情。

P2SH还实现了效率上的提升：矿工的工作是追踪那些还没有被消费掉的输

出脚本。采用 P2SH 的输出脚本会变得很小——它们只不过是个哈希值而已。所有的复杂性都被放在输入脚本中了。

3.3 比特币脚本的应用

现在我们已经明白了比特币脚本的工作机制，接下来我们看一下比特币脚本语言的一些强大应用。你立刻就能明白，比特币将指定公钥变成复杂地指定脚本，是有实际意义的。

第三方支付交易

比如，爱丽丝用比特币向鲍勃买东西，爱丽丝想货到付款，而鲍勃想见款发货。该如何处理？一个好的办法是使用**第三方支付交易**（escrow transaction）。

第三方支付交易可以用“MULTISIG”（多重签名）来轻易实现。爱丽丝并不直接付款给鲍勃，而是发起一个多重签名的交易，并规定：三个人中有两人签名之后，资金才能被支取。这三个人是爱丽丝、鲍勃与第三方仲裁员朱迪（Judy）。朱迪负责调解可能发生的纠纷。因此，爱丽丝发起了一个 2/3 的多重签名交易来付款，这个交易规定三个人中有两人签名之后，资金才能被支取。这个交易被纳入区块链后，资金被第三方监管，这三个人中的任意两个人可以决定资金的去向。现在，鲍勃觉得可以给爱丽丝发货了，通常情况下，如果爱丽丝和鲍勃都是有诚信的，鲍勃会按照爱丽丝要求发货，爱丽丝收货之后和鲍勃共同签名，把资金转给鲍勃。由于三个人中有两人签名即可完成支付，此时，由于没有任何争议，朱迪根本不需要参与。和爱丽丝直接付款给鲍勃相比，第三方支付交易并不会更复杂，只需在区块链里增加一笔交易即可。

但如果鲍勃其实并未发货，或者货物在路上被弄丢了，又或者鲍勃发的货物并不是爱丽丝想要的，这时会出现什么情况？爱丽丝觉得被骗了，所以不打算付款给鲍勃，而是想把比特币从监管账户要回来。这种情况下，爱丽丝不会签名真正完成付款，而鲍勃肯定也不会承认问题而主动放弃收款，这时，就需

要朱迪判定资金到底该转给谁。如果朱迪认为鲍勃欺骗爱丽丝，她就会跟爱丽丝一起签名，把比特币退还给爱丽丝，当然，如果她认为爱丽丝应该付款，那她就会和鲍勃一起签名，完成资金的实际支付；所以，到底该完成支付或是撤销支付，由朱迪决定。当然，这种情况也只有在发生纠纷时才出现。

绿色地址

另外一个很酷的应用叫作**绿色地址**（green addresses）。假如爱丽丝要转账给鲍勃，而鲍勃不在线（或者鲍勃在线但没有时间），所以他无法通过查看区块链的更新来确认转账是否完成。一般来说，一个交易需要获得6次确认，我们才能确信它已经确实被加到区块链中，但这需要大约一个小时。但是，想象一下，如果爱丽丝只是在鲍勃的店里买一个热狗，这么长时间才确认交易，显然是不可接受的；或者，如果鲍勃由于某些原因无法接入互联网，那他就一直无法确认交易。

为了解决这个问题，比特币采用了第三方银行的做法，实际上，“银行”可能是一个交易所，或者是其他的金融媒介。如果爱丽丝要转账给鲍勃，爱丽丝会和她的银行联系，“我要付给鲍勃这些币，你能办理吗?”银行会回答：“好的。我会从你的账号扣钱，然后从我的绿色地址转账给鲍勃。”这样，收款人就不需要实时查看区块链来确认交易。

需要注意的是，款项并不是由银行直接支付给鲍勃，实际上，部分款项，可能会通过其他地址回到银行手中。但是，由于比特币从银行控制的某个账户——我们在此所称的“绿色账户”直接转给鲍勃，而且，银行保证它不会双重支持这个比特币，如果鲍勃也相信这一点，当他看到银行签名的交易时，就可以确认自己迟早会收到这些比特币——只要区块链确认这笔交易。

请注意，这不是比特币技术系统的保证，而是现实世界中银行的保证，银行为了保护它的声誉，不会双重支持比特币。银行可以向客户证明，“我一直使用这个账户来支付，从始至终也没有发生过双重支付，我以前没有这么做，以后也不会这么做。”如果鲍勃信任银行不会进行双重支付的承诺，他就无须信任爱丽丝——他对爱丽丝本来就了解不多。

当然，如果银行出现了双重支付事件，它就会自毁长城，人们不会再信任它。实际上，有两个提供绿色地址的机构［Instawallet 和门头沟公司（Mt. Gox 的昵称，位于日本东京，是全球最大的比特币交易商）］就是由于失信而倒闭的。目前，绿色地址使用得越来越少：最初，人们认为绿色地址可以实现快速支付，而且不需要通过查看区块链来确认交易结果；但是现在，人们认为，对“银行”过分信任是有风险的。

高效小额支付（efficient micro-payments）

我们再举一个比特币脚本应用的例子。假设爱丽丝是鲍勃的客户，需要持续向鲍勃支付小额费用，例如，鲍勃是爱丽丝的手机流量提供商，根据爱丽丝每分钟使用的流量计费。但是，每分钟支付一次是不现实的：即使技术上做得到，交易手续费也让人吃不消。

我们希望可以把每分钟的费用累积起来，最后一次性支付。为了实现这种想法，爱丽丝先发起一个 MULTISIG 交易，把可能花费的最大金额转到 MULTISIG 地址，但这个交易需要爱丽丝与鲍勃两个人的签名才能生效。爱丽丝在使用流量的时候，每隔一分钟就签名一次，向鲍勃支付这分钟所产生的流量费用，然后把剩余的钱转给自己，每分钟重复一次，直到挂机为止。请注意，这些交易只有爱丽丝的签名，还没有鲍勃的签名，因此，交易还没被放进区块链里。爱丽丝挂机之后，会告诉鲍勃“我用好了，你可以切断我的服务了”，此时，爱丽丝将不再支付费用，鲍勃也将切断服务，然后在爱丽丝发送的最后一个交易里签名，把它放入区块链里。

随着每个交易付给鲍勃的币越来越多，爱丽丝的币就会越来越少。最后一个交易会一次性向鲍勃支付所有的流量费，然后把剩余的币还给爱丽丝。整个过程中，爱丽丝单独签名的交易不会进入区块链（上面没有鲍勃的签名），最后它们都会被丢弃掉。

从技术上讲，所有这些交易都是双重支付。在介绍绿色地址时，我们特别提到防止双重支付的重要性，但在本例中，我们却主动创造了大量的双重支付。实际上，如果双方都是正常运作的话，鲍勃只会在最后一个交易上签名，所以

我们在区块链上看不到中间产生的那些双重支付交易。

还有一个微妙的细节：如果鲍勃没有在最后一个交易上签名呢？他可能会说，“就让那些币待在第三方托管地址里吧。”这样一来，爱丽丝就会失去她一开始转到 MULTISIG 地址的所有比特币。但我们有一个聪明的办法来解决这个问题，那就是我们前面看到的一个代码——锁定时间。

锁定时间

为了避免上面说的这个问题，在小额支付协议开始之前，爱丽丝与鲍勃要签订一个交易，约定向爱丽丝退还所有的比特币，但是这个“退款”行为被上了锁，直到锁定时间到了为止。爱丽丝发起 MULTISIG 交易把比特币转入第三方托管之后，在向网络宣布这笔交易之前，她会从鲍勃那里要求这个退款交易。这样，如果过了 t 时间鲍勃还没有在最后一个交易上签名的话，她可以通过这个退款交易收回所有的比特币。

退款交易被锁定 t 时间是什么意思呢？还记得我们在第 3 章 3.2 节提到元数据的时候，有一个参数是“lock_time”，当时我们还没有解释。在此参数后面填上非零数值 t，这个值告诉矿工在记账的时候，要等待 t 时间之后才能把这笔交易记入区块链。这个交易在放入区块后，经过确定的区块数或者时间才生效。通过这个方式，人们可以发起一笔未来交易，当然，只有资金在未来时间点之前未被花费掉，这笔未来交易才会被执行。这在小额支付的例子里非常有效，它是爱丽丝的定心丸，能够确保在鲍勃最后没有签字的情况下她能拿回自己的比特币。

通过上面的例子，我们展示了比特币脚本可以轻易实现很多功能。我们虽然只讨论了三个例子，但其实人们研究过许多其他的功能。比如多人彩票系统，这个系统涉及一些十分复杂的多步操作协议，以及不同的锁定时间和第三方托管账户，来防止玩家作弊。还可以通过脚本语言实现多人混币，使得比特币更难被追踪。我们会在第 6 章展开讨论。

智能合约

所谓**智能合约**（smart contracts），就是那些不同于需要通过法律或者仲裁机

构来保护执行的普通合约，智能合约是比特币系统里可以用技术手段来强制执行的合约，我们已经看到，比特币有非常好的特性让我们可以用脚本、矿工和交易验证——而不是通过中心化权威机构——来实现第三方托管协议或是小额支付。

智能合约的研究目前已经非常深入，能够实现非常多很复杂的功能，但比特币脚本语言的设计也有很多缺陷，还是有很多现实需要的智能合约无法用比特币的工作控制语言来实现①。不过这里我们就不一一细谈了。

3.4 比特币的区块

现在，我们已经了解了单个交易是如何创建的，但是在第 2 章里提到，所有交易都是被打包放入区块的，为什么要这么做呢？其实这是为了性能优化，如果每一个交易都要矿工单独去达成共识，那整个系统的交易处理速度将会变得非常慢。而如果我们把大量交易组织起来放入一个区块，得到的哈希链就更短，大大提高了验证区块链数据结构的效率。

区块链（块链）非常聪明地把两个基于哈希值的数据结构结合起来：第一个数据结构是区块的哈希链，每一个区块都有一个区块头部，里面有一个哈希指针指向上一个区块。第二个数据结构是一个树状数据结构，也就是以树状结构把区块内所有交易的哈希值进行排列存储。也叫梅克尔树（请参考第 1 章），它以一种非常高效的形式把所有交易组织起来。为了证明某个交易在某个区块内，可以通过树内路径来进行搜索，而树的长度就是区块内所包含的交易数目的对数（见图 3.7）。

我们在第 2 章中提到过（在第 5 章还将继续涉及），区块头部还包含了挖矿谜题②相关的信息。还记得，区块头部的哈希函数必须以一大堆零开头才有效，

---

① 但已经有很多有意义的探索，比如以太坊等实现了图灵完备的智能合约。——译者注

② 也就是竞争记账权利问题。——译者注

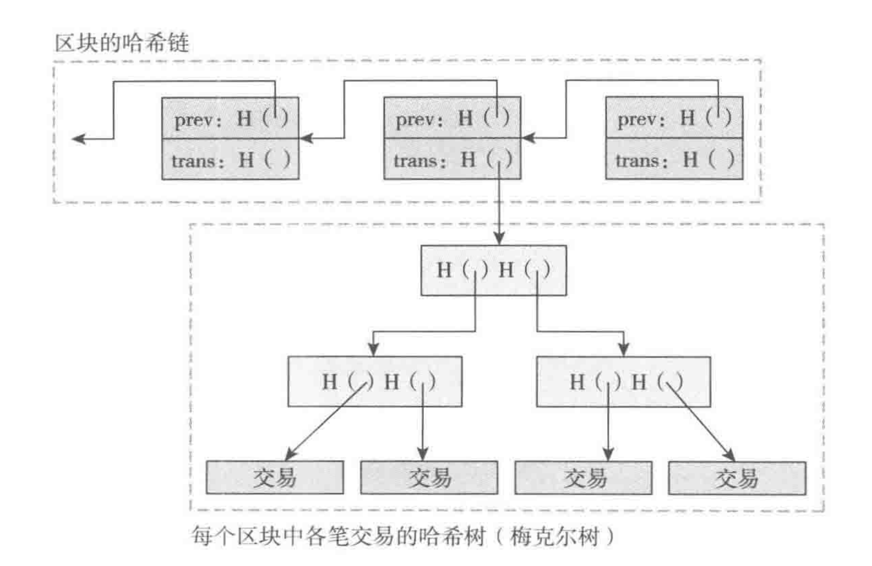


图 3.7　比特币的区块链有两个哈希结构

注：一个就是把区块联结在一起的哈希链，另一个就是区块内部的交易哈希值梅克尔树。

此外，区块头部还要包含一个矿工可以修改的“临时随机数”、一个时间戳和一个点数（点数用来表示找到这个区块的难度）。区块头部是挖矿过程中唯一哈希值化的，所以要验证一个区块的链，只要检查区块头部即可。在区块头部唯一的交易数据是交易树的树根——“mrkl_root”。

每个区块的梅克尔树上都有一个有意思的交易，叫作币基交易（见图 3.8）。这就类似于财奴币里的造币交易。这个交易创造新的比特币，它看上去像是一个普通的交易，但有几点不同：

1. 它永远只有一个单一的输入与单一的输出。

2. 这个交易并不消费之前交易输出的比特币，因此，没有指针指向“上一交易”。

3. 这个输出值目前大约是 25 个币多一点点。这个输出值就是矿工的挖矿收入。它由两部分组成：一部分是奖励的 25 个比特币（奖励在每生产 210 000 个区块——大概 4 年——后减半），另一部分是所有交易的交易手续费。

4. 还有一个特别的地方就是“币基”参数，矿工可以放任何值进去。

这里值得一提的是，当比特币的第一个区块被铸造出来的时候，该区块的

```
"in":[
      {
        "prev_out":{
          "hash":"000000.....0000000",
          "n":4294967295
        },
        "coinbase":"..."
    },
    [
"out":[
     {
       "value":"25.03371419",
       "scriptPubKey":"OPDUP OPHASH160 ... "
     }
    ]
```

图 3.8 币基交易

注：币基交易创造新的比特币，这个交易并不消费之前交易输出的比特币，因此，没有指针指向“上一交易”。币基交易的参数可以是任意数据。币基交易的价值等于区块奖励加上区块中包含的所有交易费。

币基参数提及了伦敦《泰晤士时报》的一则报道：2009 年 1 月 3 日，财政大臣拯救银行。这被看成比特币发明的政治动机，同时也很好地证明了第一个区块的打包时间是在 2009 年 1 月 3 日，上述报道出来之后。这也是矿工使用“币基”参数来支持很多比特币的不同特性。

想更好了解比特币的区块和交易结构，最好的办法是自己浏览区块链的数据，有很多网站提供数据，比如 blockchain. info，在该网站上可以看到所有交易，以及每笔交易所引用的上一笔交易。由于比特币的数据都是公开的，一些程序员已经开发出了全图形化的展现方式。

3.5 比特币网络

到这里，我们已经讨论了参与者可以发布交易，并将交易纳入区块链，这一切似乎很神奇。事实上，上述整个过程都是通过比特币网络完成的。比特币网络是一个点对点的网络，沿用了很多已有的点对点网络的理念。在比特币网络里，所有的节点都是平等的。没有等级，也没有特殊的节点，或所谓的主节点。它运

行在TCP网络上，有一个随意的拓扑结构，每个节点和其他的随机节点相连。新的节点也可以随时加入。可以试着现在就下载比特币节点软件，把你的个人电脑注册为一个节点，这个节点的权限和比特币网络里所有其他节点都是一样的。

由于随时有新的节点进入，也有旧的节点离开，所以比特币网络事实上一直在变化。并没有强制的规定节点何时明确地离开网络，只要一个节点有3个小时没有音讯，就会慢慢地被其他节点忘记。通过这个方式，网络非常缓和地处理节点下线问题。

上文提到，每个节点和其他随机节点相连，网络中并不存在一个确定的地理学意义上的拓扑结构。那么一个节点是如何加入网络的呢？当你启动一个新节点的时候，先向一个你知道的节点发送一个简单的消息。这个节点就是你的种子节点，当然，有多种不同的方法可以查找种子节点。然后你就会问你的种子节点是不是还知道其他什么节点？在链接到一个新的节点后，你可以重复这个过程许多次，最后你可以选择和哪些节点相连，这时，你就成为比特币网络里一个完全合格的节点了。这些步骤里有很多随机性，理想的情况就是你能和一些随机组的节点相连。为了加入网络成为网络节点，你只需知道一开始怎么和其中一个节点链接就行了。

那加入网络到底有什么好处？当然是为了维护区块链。当我们发起一个交易的时候，我们想让整个网络都知道。这是通过一个“泛洪”（flooding）的算法完成的［有时候我们称之为“八卦”（gossip）协议］。如果爱丽丝要转账给鲍勃，她的客户端发起一个交易，然后把这笔交易告知所有和她的客户端节点相链接的其他节点，这些节点会进行一系列核验，决定是否接受并转播这笔交易。如果核验通过，这些节点会将这笔交易信息传播给与其相连的其他节点。当节点接收到一个交易信息后，会把交易放入一个交易池，但需要注意的是，交易池里的交易还没有被打包进区块链。如果节点接收到的交易在交易池里已经存在，就不会再次把它传播出去。这样，就确保了泛洪协议会自动终结，而不是让一个交易在网络一再被传播永不停止。由于每个交易都有一个独一无二的哈希值，所以节点可以非常方便地查询某个交易是否在自己的交易池里。

节点接收到一个新交易信息时，如何核验呢？这里有四个关卡：第一个也

是最重要的一个是交易验证，也就是验证交易在当前的区块链中是有效的，节点会针对每个前序交易的输出运行核验脚本，确保脚本的返回值都为真；第二，检查是否有双重支付；第三，如前文所述，节点会检查这笔交易信息是不是已经被本节点接收过；第四，节点只会接收和传递在白名单上的标准脚本。

上述所有检查都是合理检查，所有节点很好地执行这些检查能够使网络健康、稳定地运行，但实际上并没有规则强制节点执行这些检查。虽然如此，每个节点还是有必要进行检查的——因为比特币网络是一个点对点的对等网络，任何人都可以随时加入，总有一些节点会发出双重支付，或者非标准脚本的交易，甚至彻底就是非法交易。

由于网络传递有延迟，不同的节点可能会有不同的交易池。当有双重支付攻击的时候，这个现象会变得十分有意思。假设爱丽丝想把同一个比特币支付给鲍勃与查理，于是，爱丽丝几乎同时发出两笔交易。有些节点先听到爱丽丝→鲍勃交易，有些则先听到爱丽丝→查理交易。当一个节点接收到了这两个交易当中任何一个，它就会把接收到的交易放入交易池中，之后，它听到了另一个交易，看上去像是双重支付交易，这个节点就会把它丢弃掉不再向外传播。结果就是众多的节点会对“哪一个交易应该被纳入区块链”产生分歧。这种情况被称为竞态条件[①]（race condition）。

好在对于比特币来说，这完全不是问题：打包下一个区块的矿工会打破这个僵局，他会决定哪个交易会最终打包进这个区块。如果爱丽丝→鲍勃的交易进入区块，那些听到爱丽丝→查理的节点会把爱丽丝→查理的交易从交易池里剔除，因为那是一个双重支付；而那些听到爱丽丝→鲍勃的节点也会把这个交易剔除出去，因为这笔交易已经被纳入区块链。因此，一旦这个区块被传播以后，就不再有前面说的分歧了。

由于每个节点默认保留最早接收到的交易，所以节点在网络上的位置就很重要。如果两个矛盾的交易或区块在网络上两个不同地方被发起，它们会同时向整个网络广播，节点先接收到哪个交易取决于它在网络的位置。

---

① 竞态条件也可理解为紊乱情况。——译者注

当然，这基于一个假设：不管接收到什么信息，每个节点均保留最早接收到的交易。但是比特币网络是一个对等的网络，节点并不被强制要求这么做，任何节点都有权按照其他逻辑行事，并按照所选的逻辑决定到底保留哪个交易、转播哪些交易，我们会在第 5 章的矿工奖励部分讨论这个问题。

零验证交易和费用替代策略（replace-by-fee）

在第 2 章我们讨论了零验证交易，即一旦交易在网络中广播，接收方就立即接受交易。零验证交易不是用来防止重复支付的，但由于矿工的缺省行为是把先接收到的交易放入交易池，这样，在零验证交易里就很难实现重复支付，同时，由于零验证交易非常方便，因此变得越来越普及。

自从 2013 年，矿工的缺省行为变成了“费用替代策略”，即节点在遇到有冲突的交易时，会把交易手续费更高的交易放进自己的交易池，把手续费更低的替换出去。站在矿工的角度，由于收益更高，因此也是理性的选择——至少在短期看是这样。但是这种费用替代策略却使多重支付攻击变得更容易了。

因此，费用替代策略受到了不少争议，这些争议一方面从技术层面讨论在费用替代策略中是否可以真正阻止多重支付；另一方面从哲学层面讨论比特币是不是应该要尽可能支持零验证，或直接放弃费用替代策略。我们这里就不再赘述这些讨论了很久的争议了，但最近比特币核心代码倒选用了“有选择权的”（opt-in）费用替代策略的做法，也就是交易可以标记自己是否适用费用替代策略。

上面说的是交易的传播。至于区块的传播，即矿工挖到一个矿（打包一个区块），然后将区块加入区块链，这个过程与新交易的传播过程类似，也受同样竞态条件的限制。如果两个有效的区块同时被挖到（也就是有两个矿工同时获得了记账权力时），只有其中一个区块可以进入长期共识链，哪个区块被最终纳

入长期共识链取决于其他节点选择在哪个区块上扩展区块链，未被纳入的一个即被丢弃。

核验一个区块要比核验一个交易复杂得多。除了确认区块头部，确定里面的哈希值是在可以接受的范围内，节点还必须确认区块里的每个交易。最后，一个节点往外传播的区块必须是最长的一条区块链上新加入的区块（当然，“最长的区块链”取决于节点对区块链当前状态的认识）。只有这样才可以防止区块链分叉。但就像传播交易时一样，节点同样可以执行它自己的逻辑：它可以选择传递无效的区块，也可以选择传递在共识链上更早加入的区块而不是最新加入的区块。这样就会造成一个分叉，不过这种情况是协议可以承受的。

泛洪算法（flooding algorithm）的延迟情况到底怎样呢？我们一起看一下图 3.9，这张图展示了区块被网络中不同数量的节点接收所花费的时间（秒）。三条线分别代表区块被网络中 25%、50%、75% 的节点接收到所需要的时间。可以看到，由于网络带宽的限制，比较大的区块需要 30 秒左右才能传播到大部分的节点。所以这个协议不是很有效率。在互联网上 30 秒是比较长的时间了，在比特币的设计里，简便是第一位的（简单的网络、节点可随时加入或退出），而效率是第二位的，所以在比特币网络里，一个区块可能需要经过很多节点才到达最远的节点。如果网络采取自上而下的设计，那我们就需要使任何两个节点的距离都很短。

网络大小

比特币网络大小很难测量，因为它随时都在变化，而且没有一个中央权威机构。有些人通过研究给了一些估计：往高说，每个月可能有 100 万个 IP 地址成为比特币网络的节点（也可能是临时成为节点）。往低说，大约只有 5 000 ~ 10 000 节点永远在线并处理交易。这个数字有点出乎意料得小，但是截至本书完成时，并没有证据表明永远在线的节点数量在升高或降低。

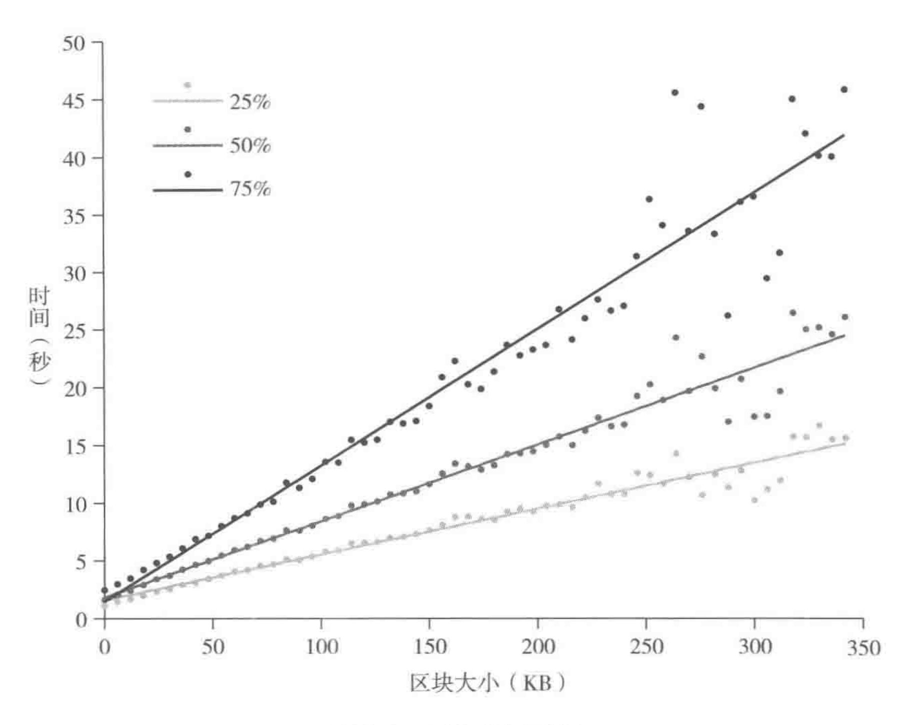


图 3.9　区块传播时间

注：展示了区块被网络中不同数量（百分比）的节点接受所花费的时间。

资料来源：Yonatan Sompolinsky 和 Aviv Zohar，加快比特币交易的传播速度（Accelerating Bitcoin's Transaction Processing，2014）。可从下述网址获得 https：//eprint. iacr. org/2013/881。数据由 Yonatan Sompolinsky 和 Aviv Zohar 授权使用

存储空间需求

完全有效的节点必须永久在线，这样才能接收到所有的交易数据。一个节点离线时间越久，当它重新连接到网络的时候，就需要越多时间来更新所有交易。这些节点还需要把完整的共识区块链都存储下来，也需要有好的网络连接，确保可以接收到所有交易并将其转播给其他节点。目前的存储空间大约要几十个 GB（见图 3. 10），一台台式机就能满足要求。

最后，完全有效节点必须维护在交易中产生的（交易的输出）、未被消费掉的比特币的完整列表，这个列表最好放在内存而非硬盘里，这样，在接收到一个交易信息的时候，节点才能快速查看、运行脚本，验证签名是否有效，然后

把交易放入交易池。到 2014 年年中，大约有 4 400 万的交易被纳入区块链，其中有 1 200 万个交易产生的比特币没有被使用。还好，这个数据不大，可以很容易地放进 1G 内存里。

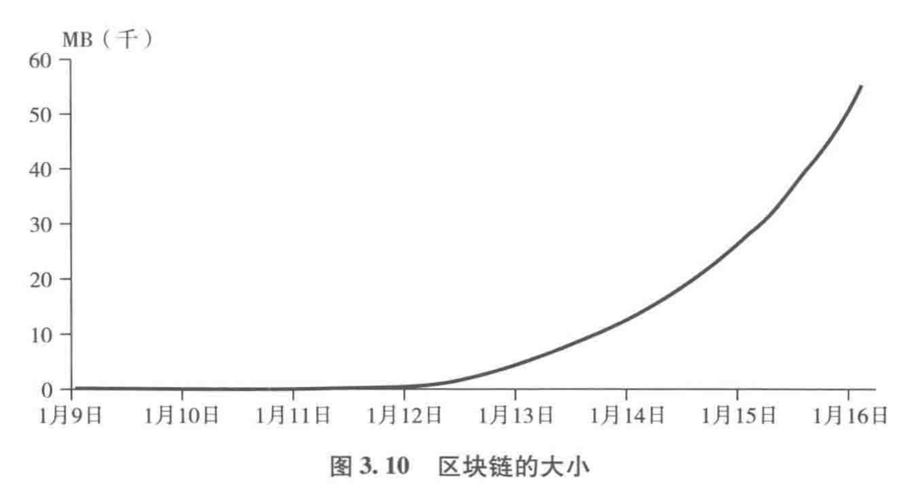


图 3. 10　区块链的大小

注：全节点必须保持整个区块链，在 2015 年年底，区块链大小在 50GB 以上。

轻量节点

除了完全有效节点之外，还有一种轻量节点（nightweight nodes），或者称为轻客户端，也叫**简单付款验证**（Simple Payment Verification，简称 SPV）客户端。事实上，在比特币系统里的大部分节点都是轻量节点。这些节点不会存储整个比特币区块链，它们只存储它们所关心的、需要进行核验的部分交易。如果你使用一个钱包软件，那里面就会有一个 SPV 节点，这个节点只会下载向你的账户付款的交易及区块头部。

一个 SPV 节点的安全等级远不如全节点。它可以核验那些很难被挖到的区块——因为它有区块头部数据，但它不能核验一个区块里所有交易记录的有效性——因为它没有所有的交易历史记录，也没有那些未被消费的比特币的列表。SPV 节点只验证那些和它们相关的交易，所以它们必须依赖那些全节点去验证网络上的其他所有交易。这虽然是一种安全性上的妥协，却不是个坏主意：轻量节点依赖全节点去处理那些比较难的工作，但当某个区块由于某些原因未被

矿工挖出来时（挖矿成本巨大），这些轻量节点也会做一些核验来确保这个区块不会被拒绝。

作为一个 SPV 节点可以节省很多资源。区块头部的大小只是整个区块链的千分之一。所以轻量节点不需要几十 G 的存储空间，只需要几十 MB 即可，即使一部智能手机也能成为比特币网络的轻量节点。

比特币是一个开源协议，比特币网络一定是由实现方式各不相同的软件系统在无缝交互。这样，即使有些软件系统有缺陷，也不至于使整个比特币网络瘫痪。比较好的现象是，人们用不同的语言不断地重新实现协议，有些人用 C + + 、有些人用 Go 语言，还有不少人用其他语言。不好的现象是，绝大部分的节点都会调用比特币官方客户端的资源库（bitcoind library），这个库是比特币核心代码开发者们用 C + + 开发的库，而且有些节点用的是过时的版本。所以，即使在同一时间，大家运行的客户端都略有不同。

3.6 限制与优化

最后我们要谈一下比特币协议的一些内在限制，以及优化的难度。在比特币 2009 年刚问世的时候，它的协议有许多内在的硬性限制，那是因为在那时没有人会想到它会发展成一个重要的国际货币。比如每个区块的平均时间、块的大小、每个区块的签名数目、切分性、比特币总量、区块奖励结构等。

比特币的总体数量与记账奖励很可能永远都不会改变，因为那样经济影响太大。矿工与投资人都在比特币现有的框架内投入巨资，如果这个框架改变了，会对他们产生巨大冲击。所以，社区基本达成共识，不管这些特性好或不好，都不应该改变。

但其他一些方面的改善可以让所有人受益——因为一些初始设计事后来看确实不太合理。其中最主要的是比特币系统的交易处理能力。每秒钟比特币网络到底可以处理多少交易？这个硬伤来自对区块大小的硬性规定，每个区块大小限定在 1MB，每个交易大约是 250 字节，所以每块最多容纳 4 000 个交易。平

均每隔10分钟，有一个矿工获得记账权利，所以每秒钟只能处理7个交易，这就是比特币网络的交易处理能力！似乎改掉这些限制只是需要改掉源代码的某些常数这么简单，实际上却并不容易，后面我们会简单分析一下原因。

比特币的交易处理能力到底属于什么水平？和前主流的一些信用卡公司相比，比特币这个处理能力实在太低了。我们可以做一下比较：维萨（Visa）平均每秒处理2 000笔交易，峰值每秒处理10 000笔交易。贝宝（PayPal）的交易处理能力比维萨弱，但峰值时每秒也能处理100笔交易。比特币无法处理这种量级的交易。

另一个限制是比特币用的密码算法。现在只有几个哈希函数算法和一个签名算法可以使用。比特币使用的签名算法是ESDSA——一种secp256k1的椭圆曲线数字签名算法（见第1章），大家担心在比特币的一生（大家都希望是很长的一生）中，这个算法可能会被攻破。哈希函数也有同样的问题，比特币使用的SHA－1也有弱点，进10年来，对SHA－1的分析也逐步取得了一些进展（尽管并不致命）。为了改变这些问题，我们不得不加强比特币的脚本语言来支持新的密码算法。

修订协议

那我们到底怎样才能修订比特币协议并引入一些新特性呢？你也许认为这很简单，只要发布一个新版本，然后更新所有的节点。但事实上非常复杂，实际中，我们根本无法假定所有的节点都会更新版本。网络里的某些节点会无法获取新版本，或无法及时获取新版本。绝大多数节点更新了协议、部分节点没有更新的后果是否严重，取决于协议更新的内容。按照产生的结果，协议修订可以分为两种类型：一种会造成硬分叉，另一种会造成软分叉。

硬分叉

通过修订协议引入新的特性，可能会使前一版本的协议失效。即运行新版协议的节点认定为有效的区块，会被运行旧版协议的节点认定为无效。而由于我们不能确保每个节点都会更新协议，我们只能假定大部分节点已经升级（新

节点），但还有部分节点没有升级（老节点），很快，最长的那个区块链分支里包含的某些区块会被老节点认定为无效区块，因此，老节点会认为其他的分支（在这个分支中，所有新节点认为有效的区块都会被排除在外）才是最长、有效的区块链分支，并一直扩展这个分支，直到它们更新了版本。

这种改变称为硬分叉，它使得原先的链分裂了。网络上的所有节点会根据其所运行的协议版本去扩展两条不同的区块链，当然，这两个分叉再也不会合并。那些老节点只要不更新版本，就被永远地排除在了另一条链之外，这是比特币社区所不能接受的。

软分叉

另一种修订是加入新的特性，让现有的核验规则更加严格。那样老的节点依然会接收所有的区块，而新的节点会拒绝一些。这样的改变叫作“软分叉”。这可以避免硬分叉所造成的永久分裂。

我们如果引入可以产生软分叉的新版协议，会有什么后果呢？运行新版协议的节点会使用一些更严格的规则，现在，假定绝大部分节点都更新了新版协议并执行新的规则（这是产生软分叉的关键，因为老节点不会执行新规则，新节点的数量要足够多才能够竞争最长的链）。这种情况下，老节点可能会挖到一些无效的区块——因为这些区块中包含一些在新规则下无法核验通过的交易，然后，老节点会知道它们核验有效的区块不被别的节点接受（即使它们并不知道原因），这使得老节点的矿工会去更新协议。而且，如果新节点用它们的区块扩展了老节点的分支，那么，老节点也会转而扩展这个分支，原因是新节点核验通过的区块，老节点也必定能核验通过。这样就没有硬分叉了，只是会有很多临时的小型分叉而已。

本章 3.2 节提到的“支付给脚本的哈希值”就是软分叉的一个经典例子。第一版比特币协议里并没有 P2SH。P2SH 之所以造成软分叉，是因为对老节点而言，一个有效的 P2SH 交易也可以核验通过——它只验证这个哈希值跟前一笔交易输出哈希值是不是一样而已，它并不知道还要进一步检验脚本是否合法。我们依赖新版节点去进行这项核验：脚本本身真的可以获取到前一个交易输出的币。

那我们到底可以通过软分叉为比特币协议添加哪些特性呢？P2SH 是成功

的，也许添加新的密码算法也可以通过软分叉实现。我们也可以通过软分叉在元数据的币基参数中添加更多的信息实现，目前，币基参数可以是任何数值，但未来我们也许可以限定币基参数的格式。已经有人提出，可以在币基参数里放入一个梅克尔树根，其中包含所有未被消费的比特币的信息。这种做法只会造成软分叉，因为老节点核验通过的区块，在新节点上可能无法核验通过。但随着区块链的延长，很快老版本就会转而去扩展最长的区块链分支。

其他的一些改变可能就会产生硬分叉了，比如在比特币里添加新的功能操作代码、改变区块大小和交易规模，甚至其他一些修复性的改动。本章 3.2 节提到过 MULTISIG 指令存在一个缺陷，它会推送给堆栈一个莫名其妙的值，要修复这个缺陷，也会产生硬分叉。这就是为什么尽管这个缺陷很烦人，但也一直没有修复，因为和硬分叉相比，保留一个缺陷还是可以忍受的。有些修订非常有意义，但目前比特币环境不太可能接受硬分叉。但许多优秀想法都在其他的竞争币中得到了测试而且成功运行，因为那些竞争币系统是从头开始建立的，硬分叉不会产生严重的后果。我们会在第 10 章进行更多的讨论。

比特币区块大小的难题

因为比特币变得越来越受欢迎，到 2016 年年初，已经开始经常发生区块被交易写满的情况，尤其是当区块在超过 10 分钟后还没有被矿工挖出来时（因为挖矿的随机性，确实有些区块在 10 分钟后还没有被挖到），这使得有些交易不得不排队等待被写进区块链。但要改变区块大小，就需要硬分叉。

究竟是否要改变，以及如何改变区块大小，在比特币社区里有热烈的讨论。这些讨论几年前就开始了，但一直进展缓慢，无法达成共识，近来讨论日趋激烈。我们在后面第 7 章会讨论比特币的社区、政治与管理。

随着区块大小问题得到共识解决方案，本章的一些细节有可能就会过时。提高比特币交易处理能力的一些技术细节很有意思，我们鼓励读者可以通过网络阅读更多的资料。

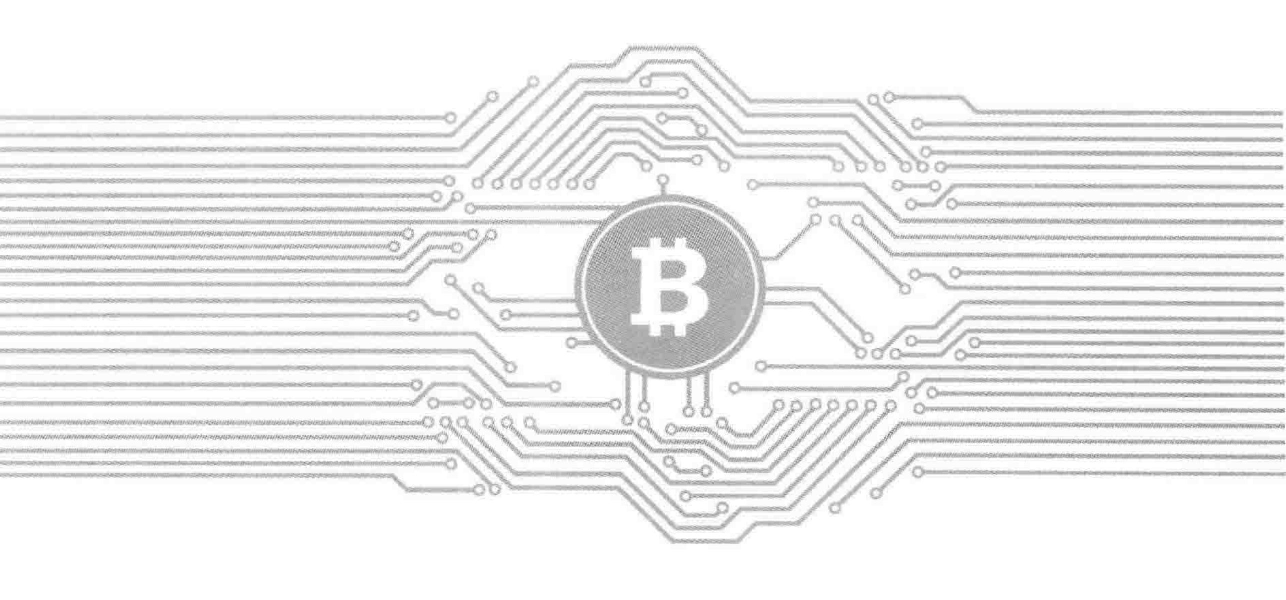
到了这里，你一定对比特币的技术机制有了一定程度的了解，也知道比特币节点是如何工作的。但是我们自身并不是一个比特币节点，你不会在大脑里运行比特币节点程序。那我们到底如何和网络进行交互，从而使比特币可以成为一种货币呢？如何让一个节点通知你交易信息呢？如何使用现金来交换比特币呢？又如何储存比特币呢？对于如何创造一种可被人们使用的货币（而不仅仅是一个软件）来说，这些问题至关重要，我们将会在下一章回答这些问题。

延伸阅读

在这一章中我们讨论了很多技术细节，你也许很难一次消化。作为本章的补充读物，你可以上网查阅一些我们讨论过的资料。网上有许多网站能让你看到区块和交易到底是什么样子的。比如有一个“区块链浏览器”，网址是：blockchain. info。

还有一本比特币开发手册也很好地讲述了一些技术细节（尤其是其中的第五、第六和第七章）：

Antonopoulos，Andreas M. *Mastering Bitcoin*：*Unlocking Digital Cryptocurrencies*. Newton，MA：O'Reilly Media，2014.



第4章

如何储存和使用比特币

本章主要讨论实际应用中如何储存和使用比特币。

4.1 简单的本地储存

让我们从最简单的储存方式开始，也就是把比特币存放在本地设备上。如我们之前所说，要使用比特币，你首先要知道一些公共的信息和一些私密的信息。公共信息就是那些记录在区块链上的内容——该比特币的识别信息、币值等。私密信息即比特币持有人——也就是你本人的私钥。你不必太担心如何储存公共信息，因为你随时都可以去调取。但是私钥则是你需要好好保管的。所以在实际运用中，储存比特币就是储存与管理你的私钥。

> 储存比特币其实就是如何保存和管理比特币私钥。

储存与管理私钥，主要有三个目标：第一是可获取性，当你要用比特币的时候，可以随时随地取用；第二是安全性，保证没有其他人可以动用你的比特币，如果有人能动用你的比特币，那他可以直接转账给自己，之后你就不再拥有这个比特币；第三是便利性，密钥管理应当是简单易行的。你可以想到，要同时做到这三点是很不容易的：

> 不同的密钥管理方法就是对上述三点（可获取性、安全性和便利性）做出权衡。

最简单的钥匙管理当然是把它们储存在你自己的本地设备上：你的个人电脑、你的手机，或你持有的、拥有的或控制的小玩意。用智能手机应用软件，按几个键你就可以支配使用你的比特币了，这么做的确非常方便。但这样做的可获取性或安全性都不是很好，如果你的设备丢失，或者你的设备死机，你需要格式化你的磁盘，或者你的文件被病毒侵蚀，你的私钥就丢失了，你的比特币也就一同丢失了。安全性方面的问题是类似的，例如有人窃取你的设备或入侵你的设备或者让你的设备中毒，将你的私钥拷贝，这样他们就可以将你所有的比特币转给他们自己了。

换言之，将私钥存储在你的本地设备，尤其是手机设备，就好比你将钱放在你的钱包里。在日常花销的时候是很方便，但你一定不想将你的毕生积蓄都带在身边，因为你不想遗失或被盗。所以一般而言，你只把一小部分信息——一小部分钱放在你的钱包里，而把你大部分钱存在其他地方。

比特币钱包软件

如果你想本地存放比特币，一般都会使用比特币钱包软件，也就是一个管理你的比特币和私钥信息并让你方便使用的一个应用软件。例如你想花相当于4.25美元的比特币在咖啡馆买杯咖啡，这个钱包应用应该很容易让你做到。比特币钱包非常有用，尤其是你需要处理一大堆地址和与其相关的密钥的时候。前面说过，制定一对公钥私钥很容易，你可以用其来匿名与保护你的个人隐私。钱包应用就是这样一个简单的接口，告诉你钱包里有多少比特币。当你要使用比特币的时候，它会处理关于密钥管理的一切技术细节，比如使用密钥或生成新的地址等。

编码解码（encoding keys）：Base58编码和二维码

要使用或是接收比特币，你需要与对方交换地址——比特币送达的地址。目前有两种主流的方式将地址加密：一种是字符串，另一种是QR（Quick Response）码[①]。

① QR码是一种简单的二维码。——译者注

为了给地址赋予一个字符串，我们把密钥的字节从二进制字符转换成 Base58 码。Base58 就是用一个包含 58 个字符的字符集来编码，这被称为 base58 记号法。为什么是 58 个字符？我们把大写小写字母都算上，然后去掉几个比较容易混淆的字母，比如大写的“O”与“0”看起来很像，就得到了 58 个字符。我们可以将加密的地址读出来，或者在需要时也能够打印出来。理想情况下，最好能避免这种手工的方式，而是采用其他方法，例如我们接下来要讨论的 QR 码。

> 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa
>
> *以上就是比特币创世块地址的 Base58 代码。*



图 4.1　QR 码

注：一个 QR 码代表着一个真实的比特币地址。请扫上面的 QR 码给我们转账一些比特币。

第二种方法是用 QR 码，一种简单的二维码。用 QR 码的好处是你可以用手机拍张照片，然后钱包应用会把 QR 码自动转换成代表比特币地址的字节。这对商店十分有用：比如一个付款机可以显示一个 QR 码，你可以用手机扫描一下，然后就可以用手机把比特币转账到付款地址。这对于手机与手机之间的转账也很有用。

虚荣地址

有些商家或个人喜欢将地址转换成一些人能够识别的字符。例如，博彩公司网站中本聪骨头（Satoshi Bones）的收款地址中就含有“骨头”（bones），如

下所示的2—6位字符（1bonesEeTcABPjLzAb1VkFgySY6Zqu3sX），当然所有的地址都是1开头的，代表支付到比特币地址的标准交易或者说是标准的比特币转账流程（pay-to-pubkey-hash）。

地址都是通过哈希计算产生的随机字符串，那么如何才能获得含有“bones”字符串的地址呢？如果中本聪骨头只是随便制定它们的地址，无法进行逆向计算哈希函数，它们无法得到相应的私钥，也无法控制地址的生成。这样的话，它们只能不停地重复生成私钥，直到私钥中包含它们希望出现的字符串。这样的地址被称为**虚荣地址**（vanity address）。这种地址事实上是可以通过工具生成的。

一般需要多少工作量能得到这样的结果呢？由于每个字符位有58种可能性，如果你想得到一个字符串中有k个字节的特殊字符，你平均需要生成58的k次方次地址，才能获得你要的结果。所以如果要生成“bones”开头的地址则要生成超过6亿个地址！这个工作现在通过一台笔记本电脑就可以完成。但是你每增加一个字符，工作量会几何级数增长。获得一个15位字符的地址需要的计算量难以想象，而且是不间断的哈希计算，这是无法实现的。

虚荣地址的加速生成

在比特币世界，如果我们将一个私钥称为x，公钥是g^x，其地址是H（g^x），即公钥的哈希值。我们不会探讨其中的细节。但是通过指数运算来生成地址显然是很慢的。

最直接的方式是挑选一个伪随机序列x，计算H（g^x），不停地生成地址，直到得到想要的结果为止。一个更快的方式是，如果使用x无法得到想要的结果，接下来就使用x+1来计算，如此反复。而不是重新挑选一个x。因为$g^{x+1}=g\,g^x$，而我们已经计算过了g^x，所以我们只需要做乘法运算而无须做指数运算，这会更快。事实上，这种方式比最直接的方式要快两个数量级以上。

4.2 热储存与冷储存

如我们所看到的，把比特币放在你的个人电脑里就像把钱放在钱包里带着，这叫“热储存”。这很方便但很不安全。而另一方面，“冷储存”是离线的，把比特币锁在其他地方。冷储存不联入互联网，是封存起来的。所以相对安全和保险，但是很显然不方便。这就像你带着一些零钱出去，但是把终生积蓄锁在保险箱里的道理一样。

要分开热储存和冷储存，你也必须要用不同的私钥，否则如果热储存被人破坏了，冷储存也会处于危险之中。你也需要把币在两边转来转去，这样两边都需要知道对方的地址或公钥。

因为冷储存是离线的，所以热储存与冷储存无法通过网络相连，但其实冷储存不需要上线就可以接收比特币——热储存端知道冷储存的地址，所以它随时可以给冷储存转账。当你觉得你的钱包里的钱太多的时候，你可以把一部分的币转到冷储存，但不需要让冷储存上线而暴露自己。当然，只要冷储存上线，就可以接收到区块链的转账信息，然后可以随意处理这些比特币。

但管理冷储存有一个小问题：一方面，为了私密性和其他考虑，我们希望使用不同的地址（这些地址有不同的密钥）收款。所以我们把比特币从热储存转到冷储存的时候，要用一个新的冷储存地址。但是由于冷储存不上线，所以热存储端必须要能找到这样的地址。

一个直接的解决方案是让冷储存一次性生成一批地址，然后把地址列表发送给热储存，热储存可以依次使用这些地址，当然，这个方法的缺陷是为了传送地址，我们不得不经常让冷储存端上线。

分层确定性钱包

一个比较有效的解决办法是使用一个分层确定性钱包（hierarchical deterministic wallet）。这个方法可以让冷储存端制造无限制的地址数量，然后通过一个短暂

的、一次性的交换，让热储存端知晓所有地址。但这需要使用密码学的技巧。

回想一下，我们在第 1 章谈到密钥生成和电子签名时，我们使用了“generateKeys”来生成一个公钥（也就是地址）和一个私钥。在分层确定性钱包里，生成密钥的方式不太一样。不同于生成一个单一地址，我们生成一个被称为“地址生成信息”的东西；我们也不只生成私钥，而是生成“私钥生成信息”。有了地址生成信息，我们就可以生成一系列地址。我们把地址生成信息和一个整数 i 作为地址生成函数的输入参数，就生成了序列里的第 i 个地址。同样，我们用私钥生成信息来生成一系列私钥。

密码学的神奇之处在于：对于每个 i 而言，第 i 个地址和第 i 个私钥相匹配——换言之，第 i 个私钥控制第 i 个地址的比特币，就好像这是用经典办法产生的。这样一来，我们就有一长串配对的公钥和密钥。

密码学的另一个技术优点是安全性——地址生成信息并不会泄露关于私钥本身的任何信息。这意味着你可以放心地把地址生成信息给任何人，他就可以用它来生成第 i 个密钥。

并不是所有的电子签名算法目前都可用于生成分层确定性密钥。比特币使用的电子签名算法 ECDSA 支持分层密钥，让我们可以使用这个技巧。即冷储存端生成任意多个密钥，热储存端生成相应的地址，见图 4. 2。

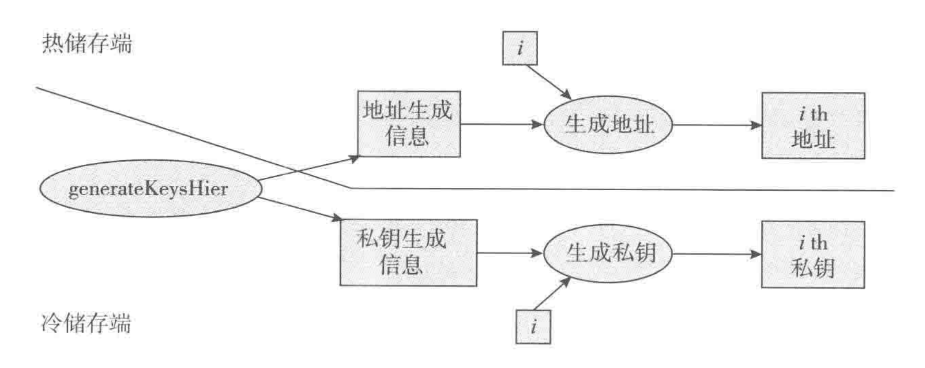


图 4. 2　分层确定性钱包示意

注：冷储存端生成和保存私钥生成信息和地址生成信息，然后将地址生成信息一次性转给热储存端。当热储存端要给冷储存端转账时，就按次序生成新的地址。冷储存端上线后，也会按顺序生成地址，然后查收相应地址收到的款项，直到某一地址没有收款位置。如果冷储存端需要向热储存端转账，它就会按顺序生成私钥序列。

ECDSA 的工作机制如下：通常一个 ECDSA 私钥是一个随机数 x，其对应的公钥是 g^x。为了生成分层确定性密钥，我们需要另外两个随机数 k 和 y。

私钥生成信息：k，x，y

第 i 个私钥：$x = y + H(k \| i)$

地址生成信息：k，g^y

第 i 个公钥：$g^{x_i} = g^{H(k \| i)} \cdot g^y$

第 i 个地址：$H(g^{x_i})$

分层确定性钱包有我们需要的所有特性：两方都可以生成公钥/私钥序列，而且这些公钥/私钥相互配对（因为与私钥 x 对应的公钥就是 g^x）。而且，这种方法还具有另外一种我们尚未提及的特性：当你向外提供这些公钥时，这些公钥之间没有联系，也就是说，别人无法断定这些公钥来自同一个钱包。稻草人方案（冷储存端生成大量的地址）也具有这种特性，但我们需要小心地保护这些地址，因为这些地址事实上并不是独立生成的。这种特性对于保护隐私和实现匿名是至关重要的，我们将在第 6 章展开讨论。

分层确定性钱包有两种不同的安全性，热储存端的安全性较低。如果热储存受到损害，那么上文提到的非相关性就不复存在，但这种情况下，私钥（以及比特币）仍然是安全的。通常，分层确定性钱包支持任意多个安全等级——这也是“分层”的由来——虽然，我们还没有讨论细节。这种安排非常有用，例如，当一家公司内部存在多种授权级别时，就需要这种特性。

现在，我们讨论一下冷储存如何保存信息（私钥或私钥生成信息）。第一种方式是将信息保存在某个设备（例如笔记本电脑、手机或平板电脑，或 U 盘）中，然后将这个设备好好保管，最好是让这些设备断开网络，并将其锁起来，这样，如果有人想盗取信息，那么他首先需要进入这些设备的保存处。

大脑钱包

第二种方法我们称之为**大脑钱包**（brain walle）。这种方式下，你通过一个密码就可以支取比特币。大脑钱包无须使用硬件、纸张或者其他长期储存介质。

大脑钱包在物理安全性较差的情况下（例如跨国出差、旅行时）非常有用。

大脑钱包的主要原理是用一个可预测的算法把一个口令变成一对公钥/私钥。例如，你可以选择一个哈希算法将口令转译成一个私钥。在给定私钥的情况下，可以用同样的方法得到私钥。进一步地，结合前文所提到的分层确定性钱包技术，你可以根据口令生成一整套地址和私钥，从而实现钱包的完整功能。

但是，如果一个黑客猜到你的口令的话，他还是可以偷走你大脑钱包里的所有私钥。在电脑安全领域里，我们通常假定黑客知道你生成密钥的步骤，黑客不知道的只是你的口令。所以黑客可以尝试使用不同的口令，生成地址，并在区块链中查看这些地址上是否还存在未被使用的比特币，一旦发现比特币，黑客就可以迅速把这些比特币转给自己。黑客可能永远都不知道（或者根本不关心）这些比特币属于谁，这类攻击也不需要入侵任何设备，猜口令不针对任何人，所以也不会留下任何痕迹。

这种方法与尝试破解电子邮箱密码的方法不同，邮件服务器通常对密码试错有一定的次数或频率限制（被称为在线猜测），但是对于大脑钱包而言，黑客可以下载一堆未被使用的比特币的地址，然后用电脑程序去慢慢地试错，黑客都不需要知道大脑钱包的地址，这被称为离线猜测或者密码破解。相应地，设置口令的难度大大增加了，又要容易记，又要不容易被猜中。一种安全的方法是使用自动程序生成一个 80 位的数字，然后将其转换成口令。

生成一个可记忆的口令

有一种简便的方法可以生成口令：从最常用的 10 000 英语词汇中，随机选择 6 个词，从而生成大致 80 位长度的字节［$6 \times \log_2$（10 000）大致等于 80］。很多人发现这个方法比随机取字母容易记忆，因为这种方法生成的口令通常是下面这样子的：

worn till alloy focusing okay reducing

earth dutch fake tired dot occasions

在实际操作中，我们可以让程序生成密钥的速度变慢（为程序加入一个延迟），这样，黑客通过试错法来破解私钥就需要花费很长的时间，这就是所谓的密钥延展（key stretching）。比如，为了使密钥生成变慢，我们可以让程序把本来很容易计算的哈希函数 SHA-256 算上 2^{20} 次，这样一来就把黑客的工作量增加了 2^{20} 倍。当然，如果太慢的话，用户在使用比特币的时候，也会计算得很慢，这也很麻烦。

如果你彻底忘记了大脑钱包的口令，钱包里的比特币就永远取不出来了。

纸钱包

第三个选择是纸钱包（见图4.3）：把密钥印在纸上，然后把纸锁在保险箱里。显然，这种方式的安全程度取决于我们所使用的纸的安全程度。纸钱包通常用两种方法为公私钥匙编码：二维码和 base58 码。就像大脑钱包一样，只需要存储少量关键信息，就可以重新建立一个纸钱包。

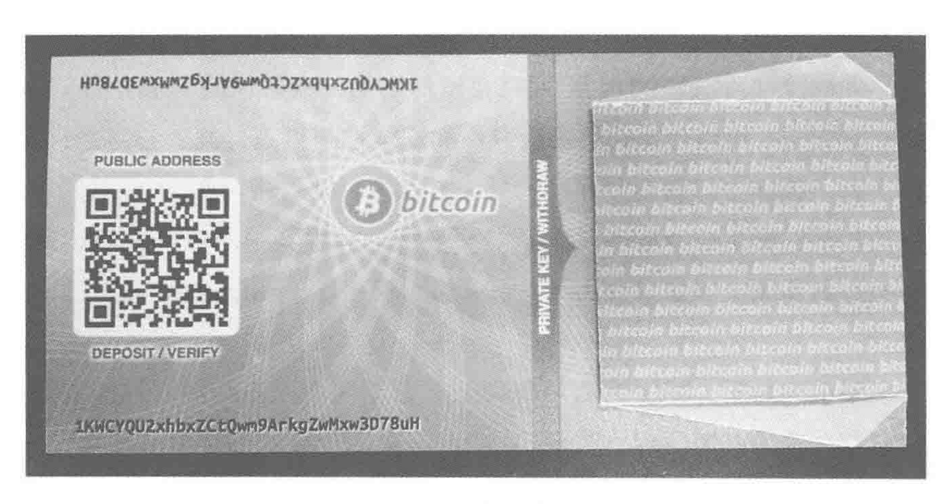


图4.3　带公钥的比特币纸钱包

注：同时使用了二维码和 base58 码加密。

防损硬件

第四种方法是使用防损硬件（tamper-resistant device），用它来保存密钥或用它来生成密钥，总之，此类设备本身不会泄露密钥或输出密钥，而只是在我们

按下设备的某个按钮或输入设备密码后显示密钥的保管状态。防损硬件的好处在于一旦设备丢失或者被盗，我们马上就能知道。而且，想要盗走密钥，必须先盗走这个设备。这和将密钥保存在电脑上是不一样的。

总而言之，用户可使用上述一个或是多个方法来保存密钥。对于热储存，尤其是存有大量比特币的热储存而言，人们愿意投入大量成本或先进的安全机制保护它们。我们将在下一章讨论这些更先进的机制。

4.3 密钥分存和密钥共享

现在我们了解了各种保存密钥的方法，但在这些方法里，我们总是把密钥保存在一个地方：要么锁在保险箱，要么保存在软件中或打印在纸上。这会造成一个问题：一毁俱毁。当然，我们可以为密钥建立备份，这可以降低密钥丢失或损坏的风险（可获取性），但同时会增加密钥被盗的风险（安全性）。那是否存在一种方法，可以使密钥的可获取性和安全性都得到提高？答案是肯定的。密码学上有一种称为“密钥分存”的技术，就可以做到这一点。

方法如下：密钥被分成 N 个片段，只要我们获得其中的 K 个片段，就可以把原密钥重新还原。但如果获得的片段数量少于 K，就无法知道关于密钥的任何信息。

要实现上述效果，把密钥简单地切分成若干片段是不行的，这样的话，每一个片段都会透露密钥的部分信息。[①] 所以，密钥分存并不是简单地切分密钥，而是将密钥转换成若干“子密钥”。

举个例子，我们设定 N = 2，K = 2，意味着我们把想要加密的密钥（原密钥）转换成两个子密钥，只有同时获得这两个子密钥才能拼出原密钥。我们把原密钥称为 S，S 是一个很大的数字（比如 128 位）。然后，我们可以随机产生

① 《鹿鼎记》里的藏宝图储存方法在现实中是不可取的，因为不需要搜集齐八旗手中的碎片，只需要有几旗的就可以猜出整个藏宝图。——译者注

另一个128位的数字R，让R作为其中的一个子密钥，那么另外一个子密钥就是S⊕R（⊕代表逻辑算符互斥，exclusive OR，或缩写成XOR，也叫异或），我们把S⊕R称为“密文”。然后，我们把子密钥R和密文S⊕R保存在两个不同的地方。单独根据子密钥R或密文都无法知晓原来密钥的任何信息，但如果我们同时得到R和S⊕R，我们可以通过异或逻辑运算得到原来的密钥。①

N和K相等时，我们总是可以这样做：对于之前的K-1个子密钥，我们可以生成N-1不同的随机数，最后一个子密钥就是原密钥与所有其他N-1个子密钥的异或。但如果N大于K的话，这个技巧就行不通了。我们需要借助其他代数方法。

图4.4中，我们是如何生成子密钥的呢？首先，我们把S标记在Y轴上(0，S)，然后经过该点画一条直线，斜率随机，接下来，我们就可以在这条线上挑一些点，要多少有多少。这样，我们就得到N个子密钥，并且K=2。

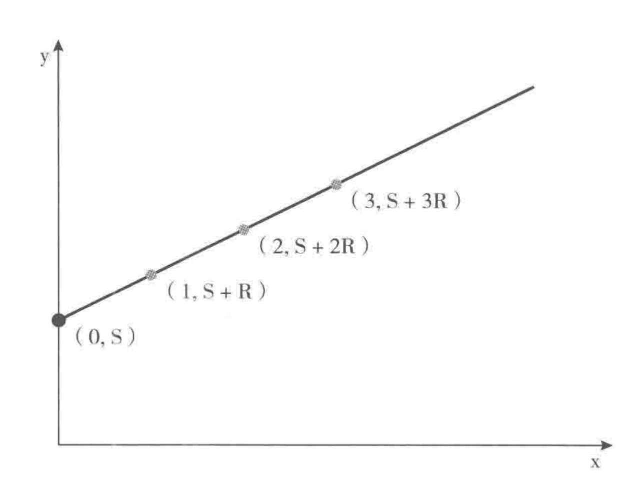


图4.4　密钥分存的几何示例（N=2）

注：S代表原密钥，被编码成一个大的整数，图中斜线的斜率随机。斜线上的点（主要是它们的Y坐标S+R，S+2R，…）代表子密钥。连接任何两个点，都可以得到S［两点连线，延长，与Y轴的交点就是S（黑点）］。若是只有一个点，又无法确定斜率（斜率随机），就无法得到S。

① 在密码学中，上文的“原密钥”通常称为“明文”，“R”称为“密钥”，S⊕R称为“密文”。——译者注

为什么我们得到两个点就可以还原原密钥呢？首先，连接两个点可以得到一条直线，这条直线和 Y 轴的交点就是 S。然而，如果只有一个点，将得不到任何信息，由于直线斜率是随机的，所以，通过该点的任何直线都有可能是我们生成子密钥所采用的那条直线，但所有直线和 Y 轴的交点都不同。

上述方法中，有一点很精妙：要让这个方法在数学上行得通，我们需要找一个足够大的素数 P 取模。P 不需要和原密钥有任何关系，只需要足够大就可以。S 的值在 0 和 P－1 之间（含 0 和 P－1）。上文中，我们说通过 S 画直线并在直线上挑选一些点，实际操作时，我们其实是生成一个介于 0 到 P－1 的随机数 R，并生成下列点：

$$x=1,\ y=(S+R) \bmod P$$

$$x=2,\ y=(S+2R) \bmod P$$

$$x=3,\ y=(S+3R) \bmod P$$

原密钥对应的坐标为 $x=0$，$y=(S+0\times R) \bmod P$，其实就 $x=0$，$y=S$。

上述方法在 K＝2，N 为任意数字的情况下都有效，例如，如果 N＝4，就是生成 4 个子密钥，并保存在 4 个不同的设备里。万一有人偷了其中的一个，他对密钥仍一无所知。即使丢了两个子密钥，你也仍然可以通过另外两个子密钥来得到原密钥。

上述方法可以进一步扩展：我们可以用任何的 K 和 N（只要保证 K＜N）来实现密钥分存。在图 4.4 中我们用一条直线进行密钥分存，在代数中，直线就是自由度为 1 的多项式。如果 K＝3，我们就要用抛物线来实现，抛物线是自由度为 2 的多项式。我们可以用表 4.1 中所示公式来表达这一递进系列。

表 4.1　密钥分存的数学原理

公式	自由度	形状	随机参数	还原原密钥所需的子密钥数量
$(S+RX) \bmod P$	1	直线	R	2
$(S+R_1X+R_2X^2) \bmod P$	2	二次曲线	R_1，R_2	3
$(S+R_1X+R_2X^2+R_3X^3) \bmod P$	3	三次曲线	R_1，R_2，R_3	4

注意，如果使用自由度为 k－1 的曲线上的若干点来进行密钥分存，那么，为了还原原密钥，至少需要得到 K 个点的数据。

数学上，拉格朗日公式表明，如果要回归一条自由度为 k－1 的曲线，需要获得至少 K 个点。最简单的例子就是，用尺子连接两个点，就可以得到一条直线。因此，如果我们将原密钥转换成 N 个子密钥，除非黑客获得了 K－1 个子密钥，否则原密钥就是安全的，换个说法，我们最多可以承受 N－K 个子密钥被泄露。

当然，密钥分存并不是比特币的专用技术。你可以将你的密码进行密钥分存，然后把子密钥告诉你的朋友，或把它们放在不同的地方。但是，实际上并不会有人真的这么做。一方面这么做不太方便，另一方面，目前市场上也有其他的安全机制，例如使用短信进行双重验证。但对于比特币而言，如果你选择本地保存密钥，那么双重验证等安全机制就不适用了，我们无法通过短信验证码的方式来控制比特币账户。当然，在线钱包则不同，我们会在下一节讨论。不过，在线钱包和本地储存的区别不大，类似的问题总是存在，只不过换了一种方式。毕竟，在线钱包的服务商在保存密钥的时候也不能只使用一种安全措施。

门限密码（threshold cryptography）

密钥分存还是有一个问题：密钥分存之后，如果我们后面要用原密钥来签名，那就需要取得子密钥，还原成原密钥，然后才能签名。这个过程有可能被黑客乘虚而入，盗取密钥。

密码学可以解决这个问题。如果子密钥储存在不同的设备中，可以以去中心化的方式还原原密钥，而不是在某台设备上完成。这种技术叫“门限签名”（threshold signature）技术。典型的例子就是使用双重安全机制的电子钱包（N＝2 且 K＝2），如果两个子密钥分别保存在个人电脑和手机上，你可以在电脑上发起付款，这时，电脑会生成一个签名片段，并发送到你的手机上，然后，手机会提示你付款信息（包括收款人、金额等），然后等待你确认。如果你确认了付款信息，这时，手机会利用它的子密钥完成整个签名，然后广播到区块链

上。万一黑客控制了你的电脑，试图把比特币转到他的账户，你根据手机上的付款信息就知道有问题了，从而不会确认这笔交易。门限密码涉及的数学细节比较复杂，此处我们不展开讨论。

门限签名

门限签名是密码学中的一项技术，将一个密钥切分成不同片段，分别储存，在交易签名时无须还原原密钥。而多重签名是比特币脚本的特性，把一个比特币账户的控制权交给多个密钥，这些密钥共同保障账户安全。门限签名和多重签名都能克服密钥单点保存的缺陷。

多重签名

还有另外一种方法可以克服密钥单点保存的缺陷，即多重签名（multisignatures），这个名词在第 3 章曾出现过。通过比特币脚本，可以直接把一个比特币账户的控制权交给多个密钥，而不是将密钥分存。这些密钥可以保存在不同的地点，并分别生成签名。当然，最终完成的交易的信息还是会保存在某台设备上，但即使黑客控制了这台设备，他所能做的也只不过阻止这个交易被广播到整个网络上去。没有其他设备参与，他无法生成出一个正当有效的多重签名。

举例来说，假设本书的作者安德鲁（Andrew）、阿尔文德（Arvind）、爱德华（Ed）、约什（Joseph）和史蒂文（Steven）是一家公司的创始人——也许我们就是依靠出售这本书的版权来创建公司——那这家公司就有许多的比特币了。我们可能会用多重签名来保护这些比特币。我们 5 个人，每人都有一对密钥，我们可以用其中的 3 个签名来保护冷储存——一笔交易需要 5 个人中至少 3 个人的签名才能完成。

这样，只要我们 5 个人在不同地方且使用不同的安全措施保存各自的密钥，那么比特币就会相当安全。黑客必须盗取我们当中 3 个人的密钥，才能盗取比特币。即便我们其中一个或两个背弃了我们，他（们）也无法卷款而逃，因为

他们还需要另一个签名。[1] 同时，如果我们其中一个遗失了密钥，其他人还是可以取出比特币，并转到新的账户，重新设置密码。总而言之，多重签名可以比较妥善地管理在冷储存端的大额比特币，任何重大事项都需要多人的参与才能实现。

上文中，我们说到，人们使用门限签名技术的原因是为了实现双重安全机制或多重安全机制，使用多重签名技术的原因是为了实现多人对共同财产实现共同控制。实际上，这两种技术都可以实现上述两种目的。

4.4　在线钱包和交易所

我们已经讨论了自己储存和管理比特币的不同方法。下面，我们将讨论如何通过他人提供的服务实现上述目的。最直接的方法是使用在线钱包。

在线钱包

在线钱包和随身带的钱包一样，只是在线钱包的信息储存在云端，你可以通过网页或手机应用来读取。2015 年年初，比较流行的在线钱包服务是比特币基地公司（Coinbase）和区块链信息公司（blockchain. info）。

从安全性的角度考虑，最关键之处在于网站不仅在你的浏览器或手机应用软件（APP）上运行代码，而且，网站还储存着你的密钥。至少，网站是能够接触到你的密钥的。通常情况下，网站使用密码来保护密钥，而密码只有你一个人知道。当然，你需要信任这个网站，相信它不会泄露你的密钥或是密码。

在线钱包的一大优点是方便。你不需要在电脑上安装任何软件就可以使用在线钱包；在手机上，你只需安装一个手机软件就可以使用钱包，而且，不需要下载区块链。在线钱包可以在各种设备上使用：无论是个人电脑还是手机，

---

[1] 用这个方法倒是可以防止银行工作人员卷款而逃，可见科技的进步确实可以改善传统行业的一些薄弱环节。——译者注

因为真正的钱包信息储存在云端。

但是，在线钱包也有安全隐患。如果网站或者是网站工作人员有恶意，那么在线钱包中的比特币就有危险。在线钱包的服务器运行着所有的代码，很容易窃取你的比特币，在线钱包服务提供商如有恶意，情况就不妙了。

通常情况下，网站或者服务提供商由训练有素的网络安全专家运行。他们比我们更专业，所以我们会认为他们帮我们保管比特币会更安全。但归根结底，前提是专家们不会故意搞破坏。

比特币交易所

要想理解比特币交易所，我们先要讨论一下传统的银行是如何运作的。你给银行一笔钱——做一笔存款，银行日后会按照你的要求把钱还给你。当然，银行并不会把你的钱一直锁在保险柜里，银行只是答应，当你提款的时候把钱给你，在这期间，银行通常会把钱用于投资。许多银行会保留一部分钱作为储备金，保证人们来提款的时候，有足够的现金。很多银行通常按存款的固定比例来留存储备金。

现在来谈比特币交易所。至少在用户使用的角度看，比特币交易所和银行很像。交易所可以办理比特币存款，日后需要用钱的时候，可以到交易所提款。你还可以把法定货币（法币）——例如美元、欧元等存到比特币交易所，交易所承诺日后会按照你的要求把钱——比特币或法币，或两者都有——还给你。也可以通过交易所办理类银行业务，例如，用比特币付款或收款。还可以通过交易所把比特币兑换成法币，或把法币兑换成比特币，交易所在该业务中通常起撮合作用，它们同时寻找愿意兑换法定货币和愿意兑换比特币的人，并安排他们作为交易对手，如果交易对手对于汇率达成一致意见，交易所就促成这笔交易。

举个例子，假设你在某交易所的账号里有 5 000 美元和 3 个比特币。你想用 580 美元/比特币的价格买两个比特币，这时交易所帮你找到交易对手并促成交易。现在，你的账号里有 5 个比特币和 3 840 美元。

值得注意的是，当你在交易所完成上述交易的时候，区块链上并不会记录

任何交易。交易所不需要在区块链里把比特币从一个地址转到另一个地址。交易所只是修改了和你的合约，交易前，它说“我们日后会还给你 5 000 美元和 3 个比特币。”交易完成后，它说“我们日后会还给你 3 840 美元和 5 个比特币。”所以，交易前后，比特币并没有真正在区块链中移动，只是你和银行的合约变化了而已。对于你的交易对手而言，也是如此。

交易所有优点也有缺点。优点之一是交易把比特币经济和法币经济相结合起来，这两种货币实现了自由转换。如果我账户里有比特币和美元，我可以随心所欲将比特币换成美元，或把美元换成比特币，这是相当方便的。

缺点就是风险。交易所面临和银行一样的风险，主要包括以下三大类风险：

三类风险

第一类风险是挤兑。挤兑就是大家同时都去银行提款。由于银行一般只保存部分存款，所以可能无法应付所有的提款要求。当银行无法兑现的谣言四起之时，大家开始恐慌，然后更多人去银行取钱，造成崩塌效应。

第二类风险是银行本身可能就是一个庞氏骗局。庞氏骗局的通常做法就是不断借新还旧，从储户吸收存款，答应日后提供一定的收益，但实际上这笔钱并没有用于投资，而是用于支付先前储户的收益，这类骗局最终必然会崩溃，使人们损失惨重。2008 年的麦道夫骗局就是庞氏骗局的最新案例。

第三类风险就是黑客入侵。有人——有的甚至就是交易所的雇员——试图入侵交易所的安全系统。由于交易所储存大量密钥，而这些密钥可以支取比特币，所以交易所需要非常小心地监控软件的安全性及其操作流程——例如，如何管理冷热储存等。如果某个环节出了差错，我们存在交易所的比特币就会被盗取。

上述风险都实际发生过。有的交易所因为挤兑而倒闭，有的交易所因为管理员的监守自盗而倒闭，也有的交易所因为黑客入侵而倒闭。实际上，统计数据并不令人乐观。2013 年的一项研究显示，40 家比特币中有 18 家由于存款到期无法兑付或其他问题而倒闭。

倒闭的交易所中，最有名的就是门头沟（Mt. Gox）。门头沟曾经是世界上最

大的比特币交易所，最后因存款到期无法兑付而宣告倒闭，许多投资者血本无归。门头沟现在还在日本与美国法院走破产清算程序，人们到现在都没有搞清楚他们的钱到底去了哪里。我们只知道一点：门头沟曾经拥有很多比特币，而现在已经一无所有了。这对于所有交易所都是一个警示。

反观传统银行业，并没有高达45%的破产率。政府的监管在其中发挥了重要的作用。政府对银行的监管主要体现在以下方面：

银行监管

政府首先要求银行有一个最低准备金要求。在美国，银行随时要保留总储蓄量3%～10%的现金来应付突发的提款请求。政府通常还会对银行的投资类别以及资金管理方法进行监管，政府要求银行的资产投向低风险资产，因为这些钱大多是储户的血汗钱，而不是银行自有的资金。

政府不仅仅对银行进行监管，还会在必要时为银行或储蓄者提供保护。首先，政府会提供储蓄保险。如果一个遵纪守法的银行破产，政府会偿还储户一部分存款。其次，政府有时候也会充当"最后借款人"角色。如果银行短期经营困难，但仍有一定的偿债能力，政府给银行提供贷款，直到银行有足够的资金周转，从而让银行渡过难关。

传统银行的监管大抵如此，但比特币交易所的监管则并非这样。比特币交易所需不需要被监管，我们会在第7章讨论。

准备金证明

比特币交易所或者其他提供比特币管理服务的机构，可以使用一种称为"准备金证明"（proof of reserve）的密码学技术来向储户证明他们留存了一部分储备金——例如，按照储蓄额的25%留存——从而消除投资人的担心。

准备金证明包括两方面的内容：首先是证明你有多少准备金。这比较容易，交易所只需发起一笔向自己转账的交易，转账的金额等于其公布的准备金金额即可。如果交易所声称留存了100 000个比特币作为准备金，那么它们会发起一笔100 000个比特币的转账交易，收款人就是交易所本身，然后向客户说明这笔

交易的有效性。之后，它们会用同一个私钥去为一条查询指令签名，这个查询指令是公正的第三方随意发出的字符串。这样就可以证明出具准备金证明的人至少知晓该私钥（即使他不是私钥的拥有者）。

我们应注意到两点：首先，严格地说，准备金证明并无法证明交易所真正拥有这些准备金，只能说明真正拥有这笔比特币的人愿意参与准备金证明的过程。换句话说，准备金证明只是证明了某人（交易所）可以控制这一笔钱，或者某人（交易所）所熟悉的人可以控制这一笔钱。其次，准备金是可能被瞒报的，一个交易所可能留存了 150 000 个比特币的准备金，但只向人们证明它留存了 100 000 个比特币的准备金。因此，准备金证明不能证明准备金的上限金额，而只能证明其下限金额，即证明某人（交易所）“至少”有多少准备金。

负债证明（proof of liabilities）

目前，交易所只证明了留存的准备金规模，为了证明准备金留存比例，还需要证明其吸收的存款规模。知道了准备金规模和存款规模，那么将这两个数相除就得到了准备金留存比例。我们接下来会展示一种方法，可以确保交易所不会瞒报存款规模（但可以多报），这样，由于交易所向人们证明了准备金“至少”是多少，存款规模“至多”是多少，这样，在计算准备金比例时，分子偏小而分母偏大，因此，我们可以得到准备金比例的保守估计。

对于比特币交易所而言，如果不考虑储户隐私的话，可以将所有的存款记录公布，即公布所有储户的姓名和金额，这样，人们就可以计算交易所的储蓄规模（即交易所的负债规模），而且，如果交易所瞒报数据，那么某些储户将发现自己并不在公布名单内或者发现自己的储蓄额少了，这时，储户就会将此曝光，因此，交易所不可能瞒报存款规模。但是，交易所可以在存款记录中加入一些虚构的客户，这样，由于公布的数据真假掺和，在一定程度上可以保护储户的隐私，只是这么做会使交易所的总负债被高估。这种情况下，只要没有收到储户投诉其储蓄被少报或漏报，人们就可以相信，交易所公布的负债规模肯定不低于实际的负债规模。

当然，以上做法是以牺牲储户隐私为代价的。实际上，我们会用梅克尔树

来证明存款规模。我们在第 1 章就说过，梅克尔树就是一棵哈希值构成的二叉树，每个指针不仅告诉我们去哪里找到一个信息，而且还告诉我们这个信息的哈希值。交易所想要证明其负债，可以先构建一棵二叉树，二叉树的每个叶节点都代表一个储户，如图 4.5 所示（当然，这里也同样需要储户来核实自己是否在这棵二叉树上），之后，我们还需要让储户可以核实交易所申明的负债规模，要实现这点，我们需要为每个节点添加一个字段（下文简称为存款金额字段），这个字段显示其最近的两个子节点的存款金额之和。

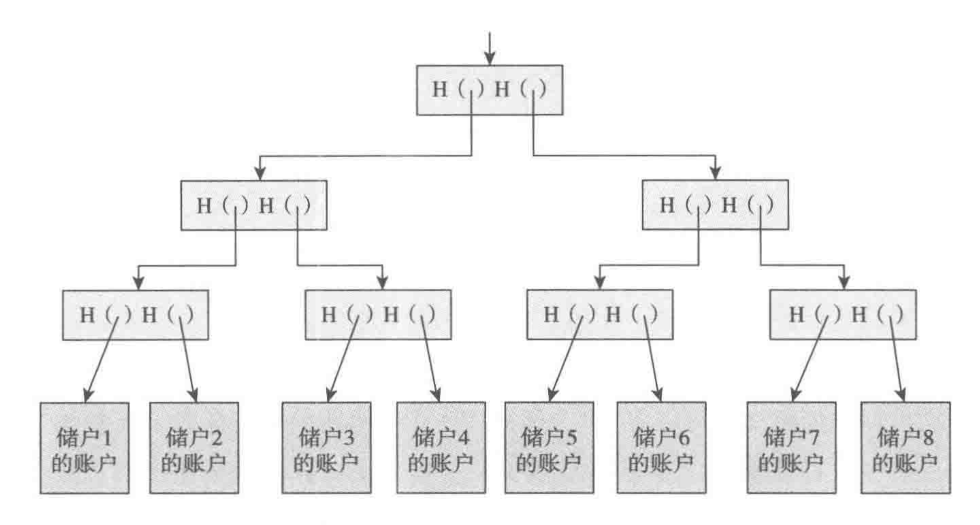


图 4.5　负债证明

注：交易所构建这样一棵梅克尔树：每个储户对应一个叶节点，每个叶节点的存款金额字段保存储户存款金额。每个节点的存款金额字段等于与其最相近的两个子节点的存款金额之和，这样，根节点的存款金额字段就代表着存款总规模。每个储户都可以要求交易所证明该储户在梅克尔树上，并且可以核实根节点所显示的总存款规模。

交易所构建完梅克尔树之后，把根节点的哈希指针和根节点的存款金额字段进行加密签名，然后在网络上广播。根节点的存款金额字段自然就是存款总规模——也就是我们最关心的数据。此外，交易所还需要声明所有储户都可以对应到叶节点上，而且所有储户的存款数据都是正确的，并且每个父节点在加总子节点的存款数据时也没有出现差错，因此，根节点的存款金额字段正确无误地说明了存款总规模。

现在，每个客户都可以向交易所索取存款证明，交易所也必须向储户出具相应证明。这种证明实际上就是一棵包含该客户所对应的节点的子树，子树应包括根节点和叶节点，如图4.6所示，之后，客户可以核实以下几点：

1. 子树根节点的哈希指针和存款金额字段，与交易所广播的值一致。

2. 从子树的根节点遍历到叶节点，每个节点对应的哈希值确实是其所指向的子节点的哈希值。

3. 每个叶节点对应的客户账号信息（客户名、账号和存款金额）都是正确无误的。

4. 每个节点的存款金额字段正好等于与其最相近的子节点的存款金额之和。

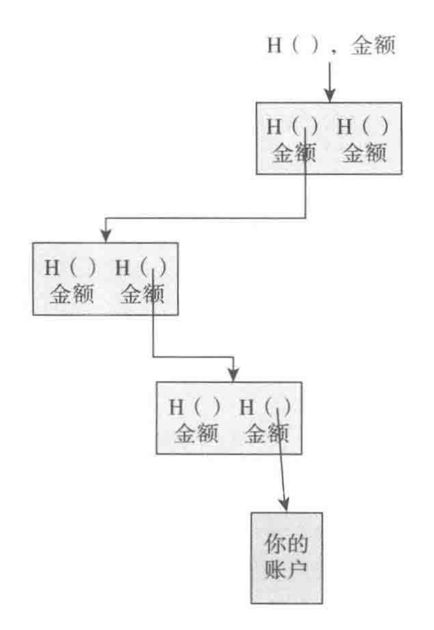


图4.6 以梅克尔树（子树）的形式提供存款证明

注：子树包含叶节点、根节点和之间的所有子节点及其兄弟节点。

上述做法的优点在于，二叉树的每个分支都会被遍历一遍，而且，总有人会核实每个节点的存款金额字段恰恰等于与其相邻的两个子节点的存款金额之和。关键的是，不同的客户获得的子树中，如果有相同的节点，那么这些节点的哈希值以及存款金额字段也必定是相同的，否则就会产生哈希碰撞（hash collision）。

我们总结一下。首先，交易所为了证明其留存了 X 比特币的准备金，发起了一笔向自己转账的交易，转账金额为 X 个比特币，并且在网络上广播这笔交易。之后，交易所证明其吸收的存款规模不少于 Y 比特币。这样，我们就知道这个交易所的准备金比例至少是 X/Y。这意味着，如果一个比特币交易所想向人们证明自己的准备金比例是 25%，可以通过上述方法让所有人都可能对此进行独立审计，而不是依靠中央权威机构来验证。

你也许发现了上述两种方法（存款证明和负债证明）泄露了很多私密信息，其中包括很多敏感信息，例如交易所使用的账户、存款准备金总规模以及交易所总负债规模，甚至是储户的个人账户余额等。实际上，交易所并不愿意公布这些信息，因此在实际应用中，存款准备金证明用得很少。

最近推出的一个被称为“准备金”（provision）的协议，也可以提供偿付能力证明，而且不需要披露总负债和总的存款准备金规模，也不需要披露正在使用的账户地址。该协议使用的加密技术更加先进，在此不深入讨论。但是这个技术又一次证明，加密技术能够帮助保护隐私。

偿付能力是一项可以执行的监管措施（比特币交易所可以自主选择遵守），其他方面的监管措施则更难执行，请参见第 7 章。

4.5 支付服务

到目前为止，我们讨论了如何存储和管理比特币。现在我们来讨论一下商户——无论是电商还是实体店——如何接受比特币付款。通常，商户接受比特币付款只是为了满足客户使用比特币支付的需求，其实商户并不愿意持有比特币，因此他们需要快速地把比特币换成法币。对于商户来说，他们希望这个过程——收款和兑换——可以尽可能简便地实现，最好是不需要了解太多技术细节。例如，不需要对他们现有的网站大动干戈，或重新购置支付设备。

商户还希望整个过程是低风险的。实际上，商户接受比特币付款可能面临多种风险。例如，使用新技术可能使他们的网站崩溃而造成损失；使用比特币

还可能存在安全风险，黑客可能攻破商户的在线钱包，或者雇员可能携比特币潜逃；最后，比特币还可能有汇率风险——比特币的汇率随时间波动很大。对一个商户而言，如果他的比萨定价是12美元，那么每卖出一张比萨，商户希望收到的钱是12美元，如果他接受比特币付款，那么他会希望比特币的价格不要出现太大的波动，他要保证所收到的比特币能兑换回来12美元。

支付服务商就在这种背景下发展起来了，可以同时满足客户和商户的需求。

站在商户的角度，客户用比特币支付的整个流程如下：

1. 商户登录支付服务网站，如图4.7所示（图来自比特币基地公司的网站界面，译者翻译），按照网站的要求，填写商品名称、数量、商品描述、收款账户等信息。



图4.7　生成比特币支付按钮的软件界面示例

注：通过支付服务商提供的网站，商户可以轻松生成一段网页代码，直接嵌入商户的现有网页即可使用。

2. 支付服务商网站会根据商户所填的内容，生产 HTML 代码，商家可以直接将代码添加到现有网页代码中，这时，网页中就会出现一个支付按钮。

3. 客户在商户网站上点击支付按钮，后台就会执行整个流程，最后商户会收到确认信息："［客户］购买了［数量］的［物品］，支付了［金额］"。

这种手动添加按钮的做法，只适用于只卖一两个物品的小网站，或用于接受捐赠的网站，对于大型的购物网站，手动复制粘贴成千上万次代码显然是不现实的。因此，支付服务商网站也提供可编程的界面来为动态页面添加支付按钮。

现在，我们来看一看，当客户使用比特币进行网购时，整个付款流程的细节是怎么样的（下面所说的步骤，正是图 4.8 描述的流程）。

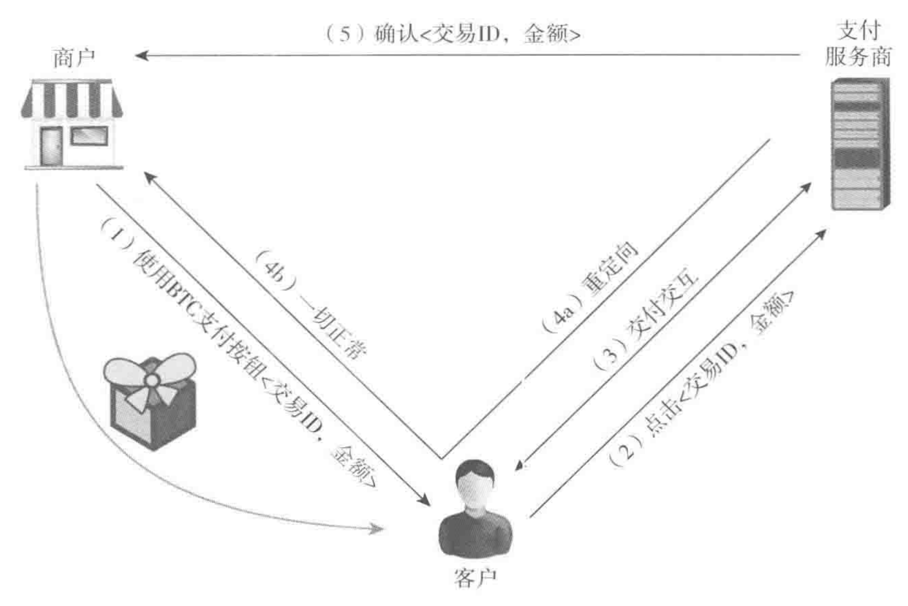


图 4.8　客户、商户和支付服务商的交互流程

1. 客户在购物网站上挑选了一个商品，当他打算付款的时候，会跳转到一个付款页面，页面上有"用比特币支付"的按钮（通过嵌入支付服务商提供的 HTML 代码实现），此外，页面上还会显示一个交易 ID（便于商户记账）和金额。

2. 如果客户想用比特币支付，就会点击对应的支付按钮，这时，网页会向支付服务商发送一个 HTTPS 请求，告诉支付服务商现在有个客户想用比特币支付，网页还会向支付服务商传送商家 ID、交易 ID、金额等数据。

3. 支付服务商收到数据之后，知道有客户——无论他是谁——想要支付一定金额的比特币，这时，网页会弹出一个窗口，提示客户付款流程，客户根据提示发起一笔转账申请，从他的钱包中转移一部分比特币给支付服务商。

4. 客户付款后，网站会重新跳转到商户页面，并提示付款流程正常。这意味着支付服务商在比特币网络上收到了这笔交易的广播，但这个交易还没有通过足够多节点的核验。从客户角度来讲，他已经完成支付，但从商户的角度讲，还需要等待支付服务商的确认才可发货。

5. 最后，支付服务商会直接向商家发送付款凭证（交易 ID 和金额），这表示支付服务商承诺会支付这笔钱给商户，这时商户才开始发货。

整个支付服务的最后一个环节是支付服务商向商户结算并付款——将相应金额的美元或其他货币直接转到商户的银行账号。结款频率可能是每日数次或每日一次，但不会每笔交易都结算一次。支付服务商会按比例收取服务费用，这就是支付服务商的盈利模式。不同的支付服务商，上述流程细节可能有所不同，但大抵相似。

简单总结一下，通过支付服务商的服务，客户可以用比特币购物，商户如期收到美元，支付服务商获得手续费，皆大欢喜。对商户来说，他只关心销售物品，收回美元或其他通行的货币，中间的一切环节由支付服务商打理：收取客户的比特币，并兑换成货币给商户。

关键的是，在整个过程中，支付服务商承担了所有风险。首先，它承担了安全风险，所以需要好的安全措施来管理比特币；其次，它承担了汇率风险，它收取比特币然后支付美元，如果美元兑比特币的汇率波动太大，支付服务商可能会遭受损失，但如果汇率往有利的方向波动，也可能大赚一笔。支付服务商的商业模式决定了它必须承担风险。

需要注意的是，支付服务商的资金流动很大，它收取大量的比特币，然后付出大量的美元。因此，支付服务商自然就成为交易所的活跃成员——通过交

易所才能实现法定货币与比特币的通兑。对于支付服务商而言，不仅需要考虑比特币的汇率问题，也要考虑如何进行巨额兑换。当然，如果一个支付服务商解决了这些问题，就可以解决客户想付比特币而商户想收美元的矛盾，为此，支付服务商从每笔交易中收取的手续费有可能使其实现相当可观的利润。

4.6 交易费

前面的章节中已经提到过交易费，以后的章节也会陆续提到。本章主要讨论在比特币系统中，交易费是如何设定的。

当一笔交易被纳入比特币区块链的时候，就可能支付了一笔交易费。前面章节提到，交易费是比特币交易中输入金额（付款方支付的比特币）和输出金额（收款方收取的比特币）之间的差额——输入金额必须不少于输出金额，否则交易无法完成；如果交易的输入金额大于输出金额，那么这个差额就是交易费。交易费由将该交易打包进某个区块的矿工获得。

交易费的经济学意义非常有趣，也非常复杂。而且交易费的设定细节随时间变化而不断变化，我们在这里把时间框定在 2015 年年初，讨论交易费在那个时点是如何设定并运作的。我们也会在本节末尾简单叙述一下当前的情况。

交易费为什么存在？原因是在比特币网络中传播你的交易信息是需要成本的——每个节点传播交易信息，最后由一个矿工把这笔交易打包进一个区块，这些都有代价。举个例子，如果一个矿工将你的交易打包进他的区块，那么这个区块就会比其他区块大一点点，也会花费更多的时间传输到其他节点，这样，这个区块变成孤块的可能性就会提高，原因是在这一小段时间中，别的矿工也许刚好完成了区块打包。

不论是对于节点的传播还是对于矿工而言，确认你的交易都需要花费代价，交易费便用来补偿矿工处理交易所付出的代价。节点通常并不收取费用，当然，节点的运行成本也比矿工的运行成本低许多。交易费的金额可以由发起交易的人自由设定，也可以不设定交易费。通常来讲，如果你支付了较高的交易费，

则交易将被更快、更可靠地传播和记录。

为了说明矿工如何设定交易费用，我们现在来看一下默认的交易费政策。但需要注意：首先，下文叙述的交易费政策基于2015年发布的现行第0.10.0版本，在后续版本中，交易费政策可能不同；其次，设置默认交易费的动机是为了防止区块链被大量小额交易所“污染”，而并不是想准确地评估矿工处理交易的成本。

当然，矿工无须遵守默认的交易费政策。在2015年，交易费在矿工收入中的占比不到1%，因此，大部分矿工是遵循默认交易费政策的。但随着挖矿奖励的降低，交易费在矿工收入中的占比会越来越高，我们可以预测，将会有越来越多的矿工不再遵守默认的交易费政策。

现在大部分矿工所默认的交易费是这样计算的：首先，如果交易满足以下三个条件，那就不需要支付交易费：

1. 交易小于1 000个字节。
2. 所有输出为0.01BTC或更大。
3. 优先权足够高。

优先权是这样定义的：（所有输入的账龄的总和×输入金额）/交易规模。也就是说，先明确交易的输入所对应的上一笔交易，把每笔交易的账龄（交易完成到现在所经过的时间）及输入金额相乘，然后把乘积相加，就得到了优先权数据。注意，某笔交易的输出越长时间没有被消费掉，账龄就越高，那么，它被支付时，交易的优先权就越大。

如果一笔交易满足了上述三点要求，这笔交易会被传播，最后会被纳入区块链，这个过程是免费的。否则，交易就会被收取费用，当前默认标准是：每1 000个字节需要支付0.0001BTC，在2015年，这相当于每1 000个字节花费1美分的交易费，一笔交易通常包括：输入通常是148个字节，每个输出通常是34个字节，其他信息10个字节。如果一笔交易有两个输入和两个输出，那么这笔交易的大小是400个字节。

如果一笔交易没有满足上述要求，它也有可能会被纳入区块链，但如果你想要一笔交易被更快、更有保证地纳入区块链，那么就需要支付一笔标准的费

用。因此，大部分钱包软件和支付服务商在它们的支付流程中，都包含了标准的交易费用，在日常比特币交易过程中，你会发现你支付了一定的交易费用。

当前，大多数的矿工会强制要求交易必须包含交易费，这意味着他们不会去处理那些没有付交易费的交易（或者最后处理）。当然也有些矿工没有这么做，他们愿意接受很少的交易费用甚至免费。

4.7 货币兑换市场

这里所说的货币兑换市场是用比特币来交易美元或是欧元。虽然前面谈到的支付服务也能做到这一点，但这里讨论的是整个货币兑换市场，包括市场规模、分布、如何运转，以及与这个市场相关的经济学原理。

首先，比特币兑换市场的运作和法币兑换市场的运作很相似。货币兑换价格的涨跌取决于人们买入美元或欧元的需求。在比特币世界里，有不少网站（例如 bitcoincharts. com）可以告诉你比特币与不同货币之间的汇率。

在这个网站上，可以看到有很多交易正在进行，汇率也持续在发生波动。比特币市场具有高度的流动性，你可以在很多场所兑换比特币。2015 年 3 月，Bitfinex（最大的比特币和美元兑换交易所）的每日交易量大约是 70 000 比特币，或是 2 100 万美元。

当然，人们也通过当面交易的方式兑换比特币。有许多网站提供这种服务，举个例子，在 localbitcoins. com，你可以告诉别人你的位置，然后告诉大家想以什么样的价格购买多少比特币，之后，网站会告诉你当地有多少人愿意在约定的地点出售比特币，还会告诉你他们出售比特币的数量和价格。这样，你就可以联系他们，约在一个咖啡馆或公园，或者其他什么地方进行交易——支付美元购买比特币。对于小的交易，在交易完成之后，你们只需等待一小段时间，交易便可以在区块链上被确认。

交易变得越来越频繁，人们开始定期在固定地点碰头，进行比特币交易。例如，你可以在固定的地点，去某个公园、街角或者咖啡店，大家都在那里聚

会，进行比特币的买卖交易。有很多人喜欢这样的线下交易，与交易所在线交易相比，线下交易可以保护用户的个人隐私，而根据银行的监管要求，人们在交易所开户需要提供身份证明，无法实现匿名。我们将在第 7 章中对此展开讨论。

供给和需求

就像其他市场一样，比特币交易所撮合买家与卖家。交易所是一个相当大的市场，每天的交易量都在数百万美元的规模。当然，它还比不上纽交所或是能和美元/欧元外汇市场相提并论，但它也成了一定的气候，而且比特币兑换也形成了公允市场价格。一个交易员想在交易所买卖比特币，总是可以找到交易对手——至少在交易额不太巨大的情况下如此。

和任何一个其他流动的市场相似，比特币市场中的公允价格是由供给和需求决定的。供给是指比特币的供给，即可能被出售的比特币的量；需求是指人们对比特币的需求，他们持有美元，想购买比特币。通过市场的供给和需求机制，比特币的价格会被设定为一个价格，在这个价格下，比特币的供给和需求刚好相等。我们稍微讨论一下细节。

比特币的供给是多少呢？供给就是指人们可能买到的比特币数量，也就是整个市场中正在流通的比特币数量，这个数值是固定的，在 2015 年年末，这个数值是 1 500 万，根据比特币的设计，比特币数量将缓慢上升，最终达到 2 100 万的上限。

在考虑比特币供给的时候，也可以将活期存款考虑在内。如果人们在交易所存入比特币，而交易所并没有全额提取准备金，那么实际上，活期存款总量实际上超出了交易所实际留存的比特币数量。

把活期存款算在比特币总供应量里是否正确，取决于我们如何定义总供应量。如果在我们所分析的市场中，活期存款可以被出售，例如，如果人们可以用比特币存款兑换美元，而且，人们要求提取活期存款时可以要求交易所支付美元，则活期存款应该被记入比特币总供应量。

值得注意的是，通常情况下，当经济学家谈到法定货币的供应量的时候，

他们其实也包括了活期存款，而不仅仅计算市场上流通的货币（即流动的纸币和硬币），原因是人们也使用活期存款来购买商品。所以，当我们说一个市场上比特币的供应量固定在 1 500 万或者最终增加到 2 100 万时，我们需要把那些可以等同于现金使用的活期存款也考虑进来，因此，比特币的供应量和某些比特币专家所声称的数量可能不同。我们要针对特定的市场情况来讨论比特币的供应量到底是多少。所以，我们后面讨论比特币供应量的时候，实际上都是针对我们所分析的特定市场而言的。

现在我们再来看一看需求。比特币的需求可以分为两类：一类是将比特币作为支付中介，另一类则是投资需求。

第一，我们讨论作为支付中介的比特币。想象一下，爱丽丝想从鲍勃处购买某个商品，而且希望用比特币来支付（假定爱丽丝和鲍勃其实也可以用美元支付，但他们发现用比特币支付更方便）。我们再假定爱丽丝和鲍勃都不想长期持有比特币，所以，爱丽丝首先会用美元兑换一些比特币用于支付，鲍勃收到比特币之后，会再将比特币兑换成美元。这个例子的关键在于，用于支付的这部分比特币实际上短暂地退出了比特币流通体系。这就产生了比特币的需求。

第二个需求是投资需求。有人想购买比特币并长期持有，等价格上涨之后卖掉。当人们购买比特币并长期持有时，这些比特币也就不再流通。当比特币价格很低的时候，大家都会想买进比特币用来投资；而当价格很高的时候，需求则不会很高。

一个简单的市场行为模型

我们现在做一个简单的经济学建模来理解这些市场行为。我们不会推演整个模型（尽管推演过程很有意思），而是将着重讨论比特币兑换需求对比特币价格的影响。

我们先假设一些参数。T 是指市场中所有参与者用比特币进行支付的总交易量，该数值用每秒钟发生的交易量（美元）来计量，我们假定人们在进行支付的时候，也总是想着比特币的美元价值，这样会简化问题。这样，我们就可以用美元来衡量每秒钟的总交易量。D 是指比特币用于支付时，暂时退出流通

体系的这段时间长度——从付款者买入比特币开始到收款人将比特币兑换回市场为止，以秒计算。S 是人们可以买到的比特币总量，等于比特币总量——目前是 1 500 万左右（未来会增加到 2 100 万左右）——减去人们打算长期持有的比特币数量，也就是在市场上流通的、可以随时买卖的比特币总量。最后，P 是比特币对美元的价格。

现在我们可以做一些计算。首先，我们要计算一下每秒钟有多少比特币可以被用来做交易。在 D 秒内，市面上有 S 个比特可以用于交易，所以每秒钟有 S/D 的比特币重新进入流通体系，可以用于支付。这是供应侧的数据。

在需求侧（指每秒钟所需的用于支付交易的比特币数量），我们总共的支付交易规模是 T 美元，而每 1 美元对应的交易，我们需要 1/P 个比特币来完成。所以，T/P 就是我们每秒钟所需的用于支付交易的比特币数量。

在特定的某秒钟内，供应是 S/D ，需求是 T/P。和其他市场一样，价格会根据供需关系达到平衡。如果供应大于需求，有些比特币就卖不出去，出售的一方不得不降价出售。同样地，根据我们 T/P 的需求公式，当价格下降后，需求增加，供应与需求会再次达到平衡。

另一个方面，如果供应小于需求，这意味着有些人想购买比特币用于支付，但是买不到，这些人就必须出更高的价格来购买，而当价格升高后，需求就会下降，供应与需求也会再次达到平衡。所以我们可以得到以下公式：

$$S/D = T/P$$

从中我们可以推算出价格的公式：

$$P = \frac{TD}{S}$$

这个公式说明什么？我们可以将其更加简化：我们假设 D（也就是比特币用于支付时，暂时退出流通体系的这段时间长度）不变，人们可以买到的比特币总量 S 也不变，或至少变化的速度很缓慢。这意味着价格 P 和需求 T 是成正比的。所以，如果需求翻倍，比特币价格也会翻倍。我们可以根据比特币的实际价格以及我们估计的需求量做图，看看价格和需求的关系是否真的和我们所

预测的一样。经济学家们已经证实了这一关系。

请注意，S（人们可以买到的比特币总量）并不包含那些用于投资而被长期持有的比特币。所以如果越来越多人购买比特币用于长期投资，则 S 会下降，根据上面的公式，P 会上涨。这很容易理解，当持有比特币作为长期投资的人增多，用于支付中介的比特币的价格自然上涨。

这并不是一个完整的市场模型。一个真正的市场模型还必须考虑投资者的心理活动，如果投资者相信比特币价格会上涨，他们对比特币的需求就升高。这就是我们所说的投资者预期，加入投资者预期后，模型会变得很复杂，我们在这里不展开讨论。

总之，存在比特币和美元、比特币和其他货币的兑换市场。这些市场有足够的流动性，你可以兑换或买卖比特币，这种兑换很可靠，当然价钱会有波动。我们可以用经济学模型来理解供给和需求对市场的影响，如果我们有办法估计一些不可知的信息，比如说未来人们需要多少比特币用于支付，我们就可以预测市场。经济学模型很重要也很有用，现在也有很多人从事更细节的研究，但详细的经济学模型并不在本书的讨论范围。

延伸阅读

比特币的安全机制和银行的安全机制类似，但也有些重要的差异。Ross Anderson 的安全机制教材一书中的第 10 章“*Banking and bookkeeping*”，非常值得一读，教材可以在网络上免费获得：

Anderson，Ross. *Security Engineering*. Hoboken，NJ：John Wiley & Sons，2008.

分析比特币交易所为何倒闭的书籍，建议阅读：

Moore，Tyler，and Nicolas Christin. “Beware the Middleman：Empirical Analysis of Bitcoin-Exchange Risk.” In *Financial Cryptography and Data Security*. Heidelberg：Springer 2013.

Adi Shamir 关于密码分存的书籍：

Shamir, Adi. "How to Sharea Secret." *Communications of the ACM* 22 (11), 1979.

讨论 Provisions（无须泄露储户隐私的偿付能力证明协议）的相关书籍：

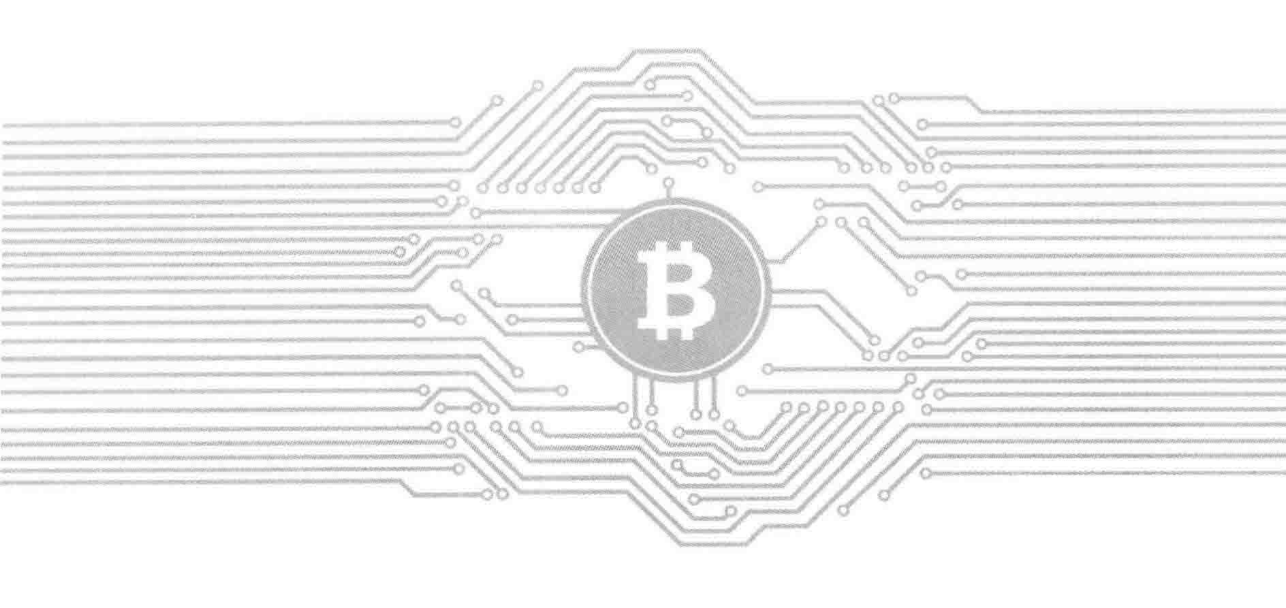
Dagher, Gaby and Benedikt Bünz, Joseph Bonneau, Jeremy Clark, and Dan Boneh. "Provisions: Privacy-Preserving Proofs of Solvency for Bitcoin Exchanges." In *Proceedings of the ACM Conference on Computer and Communications Security*. New York: ACM Press, 2015.

一个密码很难又好记又安全——现在的密码破解技术越来越巧妙、有效，下面的文章有相关示例：

Weir, Matt, Sudhir Aggarwal, Breno De Medeiros, and Bill Glodek. "Password Cracking Using Probabilistic Context-Free Grammars. Presented at the 2009 IEEE Symposium on Security and Privacy, Oakland, CA, 2009.

关于 2014 年比特币交易费的调查报告：

Möser, Malte and Rainer Böhme. "Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees." Presented at the Second Workshop on Bitcoin Research, Puerto Rico, 2015.



第5章

比特币挖矿

BITCOIN
AND
CRYPTOCURRENCY
TECHNOLOGIES
A Comprehensive Introduction

这一章节我们将着重讨论比特币挖矿（bitcoin mining）。前面我们已经讨论了比特币是如何依赖这些矿工们而运行的——他们查证交易记录，制造和储存所有的区块，并对被写入区块链的区块达成共识。我们还知道矿工们会从中得到一些奖励，但还是有些悬而未决的问题：这些矿工都是谁？他们是如何进入这个行业的？他们是怎么运作的？他们的商业模式是什么？他们对环境造成什么影响？我们将在这一章节回答这些问题。

5.1 比特币矿工的任务

你若是想加入比特币挖矿行列，我们不会极力劝阻你，但会提醒你比特币挖矿潮很像当年的淘金热。历史上的淘金热充斥着各种年轻的淘金者下海淘金发财的故事，并且不可避免地，许多人最终失去了一切。在这些确实经历了千辛万苦的人们当中，也只有少数人变得富有。我们将在本章中了解到，和传统淘金以及其他快速致富途径一样，比特币挖矿也面临着类似的挑战和风险。我们首先要看一下技术细节。要成为比特币矿工，你必须加入比特币网络并与其他节点相联。建立链接之后，还有六个任务要完成：

1. 监听交易广播。监听网络上的交易广播，然后验证它们的签名是正当有效的，交易输出没有被重复支付。

2. 维护区块链网络和监听新的区块。必须先维护区块链。为了做到这一点，

一开始你可以要求其他节点把区块链上的历史记录（在你加入区块链网络之前的）同步过来。然后，监听那些被广播到网络上的新的区块。你的任务是验证你收到的每个区块，这里的验证是指保证区块里的每笔交易都是有效的，而且这个区块包含了一个有效的随机数。我们会在本章后面谈到验证随机数的技术细节。

3. 组装一个备选区块。一旦拥有最新的全部区块链数据备份，你就可以开始制造你自己的区块了。要做到这一点，你要把所监听到的交易进行组合并放进一个新的区块，然后把该新区块排在整条链中最新的区块的后面。你必须保证你建立的新区块里的每笔交易都是正当有效的。

4. 找到一个让你的区块有效的随机数。这一步的工作量最大，也是矿工工作中最难的一个环节。我们后面会谈到细节。

5. 希望你的区块被全网接受。即使你找到了一个区块，也不能保证该区块会成为共识链（concensus chain）的一部分。这需要有点运气，希望其他的矿工接受你的区块，然后从该区块开始继续接龙下去，而不是从你的竞争对手发现的区块开始。[①]

6. 利润。如果所有其他矿工接受了你的区块，那你就能获取利润。在 2015 年，一个区块的奖励是 25 个比特币，大约在 10 000 美元左右。此外，如果在该区块里的任何交易都有交易费，所有交易费也会为矿工所有。到目前为止，交易费作为额外收入，相对来说还比较低，大概是一个区块默认奖励的 1%。

我们可以把矿工的任务分成两类：第一类任务是验证交易和区块，这是比特币网络赖以生存和运转的基础。这些任务也是比特币协议需要矿工的首要原因。第二类任务是和其他矿工竞争，争取可以找到区块并因此获益。这些任务并不是比特币网络存在所必需的，而是为了鼓励矿工去完成第一类任务而设置的。当然，这两类任务都是使比特币成为一个数字货币的必要条件，因为矿工必须获得奖励才会去完成这些重要的任务。

---

① 如果下一个区块始于其他人发现的有效区块，你发现的这个区块就会变成无效区块而被丢弃。——译者注

寻找有效区块

现在回到如何找到一个使区块有效的随机数的问题上。在第 3 章中我们讨论过，区块链主要有两层基于哈希函数的结构。第一层在是区块链上，每个区块的头部都有一个指针指向其前一个区块，第二层是在每一个区块里，包括所有交易的梅克尔树。

作为矿工，首先需要从你的交易池中选出一系列有效的交易并且编译成梅克尔树。当然，只要不超过每个区块随机数的交易上限，你可以选择编译的交易数量。然后，组装出一个新的区块，让它的头部指向区块链上的前一个区块。新区块的头部，有一个 32 位的随机数区域。你需要尝试不同的临时随机数，直到该随机数能使整个区块的哈希值小于目标值。这个目标值一般体现为以零开始的特定位数的数值。作为一名矿工，你可能使随机数从 0 开始，每次增加数值 1，直到该随机数能使区块有效为止，如图 5.1 所示。

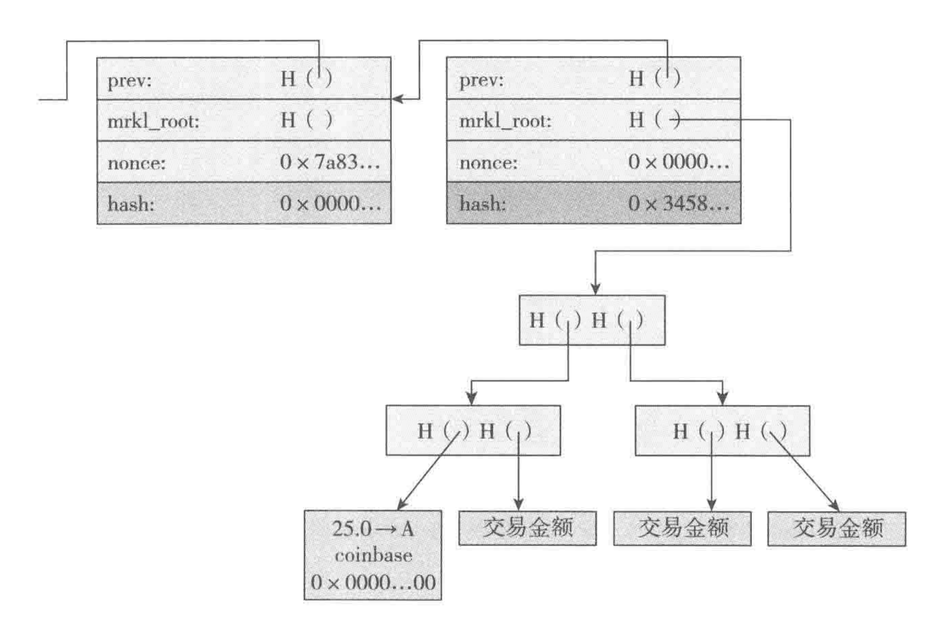


图 5.1　寻找有效区块

注：在这个例子中，矿工尝试了一串都是零的临时随机数，但没有产生有效的哈希值，所以矿工继续尝试其他不同的临时随机数。

在大多数情况下，随机数试过所有32位可能的取值后，仍然不能产生一个有效的哈希值，这时候你必须做出更多的改变。注意，图5.1中币基（coinbase）还有一个随机数可以改动。当你遍历区块头部随机数所有可能的取值后，可以改变币基里的随机数，比如加1，然后可以重新改变区块头部随机数来寻找有效的哈希值。

当改变币基里的随机数后，整个梅克尔树上交易的哈希值都会改变（见图5.2），因为币基值的改变会向上传递，所以改变币基的随机数值比改变头部随机数值的代价要大很多。正因为如此，矿工大部分时间只改动头部的随机数，只有在遍历头部2^{32}个随机数值且还没有找到一个有效区块时，才改动币基的随机数。

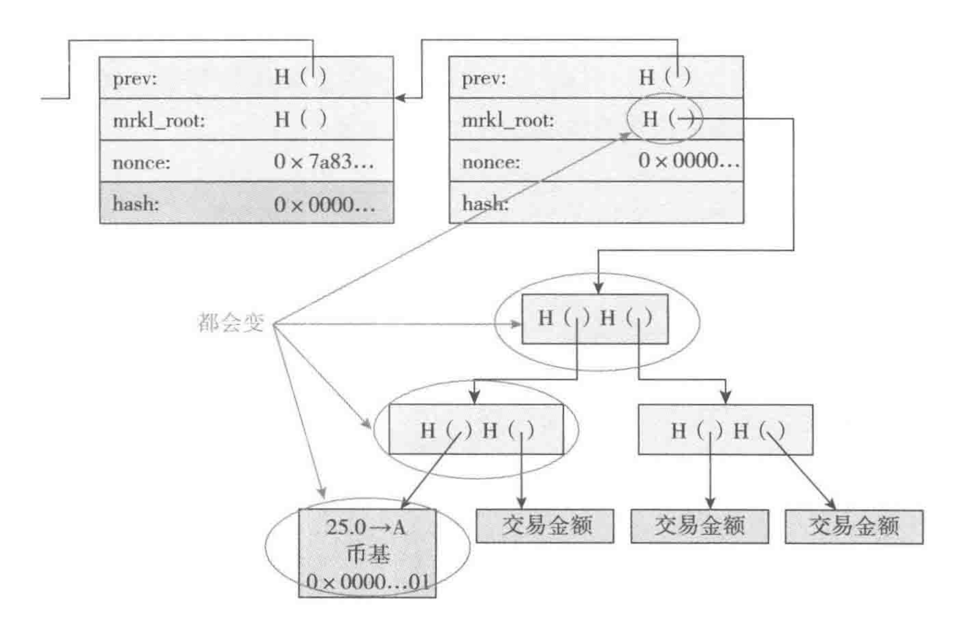


图5.2　改变临时随机数

注：改变币基里的临时随机数，整个梅克尔树的哈希值都会因此而改变。

你所尝试的绝大多数临时随机数都不会成功，但若能够坚持足够长的时间，你总能找到一对正确的临时随机数组合——头部随机数与币基随机数，用来产生一个符合哈希值要求的新区块。找到后要立即宣布，就有希望得到相应的区块奖励。

每个人都在运算同一个谜题吗?

你可能会想，如果每个矿工都在临时随机数值上逐步加1，岂不是大家都在按照同一个运算模式解同样的谜？岂不是最快的一个总赢得竞争吗？不会的。首先，矿工们不太可能在完全相同的一个区块上进行运算，因为每个矿工都会把或多或少不同的交易用不同的次序来放进区块内。但更关键的是，就算两个不同的矿工所组建的区块里包括了一模一样的交易，这两个区块的哈希值还是会不同。请记住在币基交易里，矿工会写自己的地址接收新铸币。这个地址本身的区别会沿着梅克尔树往上传递直达树根，导致整棵树上的哈希函数值不同，从而保证了没有两个矿工的区块是一样的。除非两个矿工共享公开密钥。这种情况只有可能两个矿工同在一个矿池（我们以后会讨论）。同在一个矿池的矿工会互相通信，确保使用不同的币基临时随机数以避免重复工作。

找到一个有效区块到底有多难？到2015年年底，这个挖矿的难度目标区域值（用16进制来表示）为：

0000000000000000172EC0000000000000000000000000000000000000000000

所以任何有效区块的哈希值必须低于这个值。换句话说，大约2^{68}个临时随机数里只有不到一个可以成功，这是一个非常巨大的数值。一个粗略的估计，它比全球人口总和的平方还要大。也就是说，如果地球上的每个人都是一个包含7亿人口的独立星球，那么总人口将会是2^{65}。

决定难度

每挖出2 016个区块，挖矿难度会改变一次，这个周期大约是两个星期。难度的改变是根据上2 016个区块的挖矿效率来决定的。用下列公式来表达：

$$\text{下一个难度} = \frac{\text{上一个难度} \times 2\,016 \times 10\ \text{分钟}}{\text{产生上}2\,016\text{个区块所花费的时间}}$$

注意，2 016 × 10 分钟就是两周，也就是说，如果产生一个区块需要 10 分钟，那么产生 2 016 个区块就需要两周时间。所以这个公式的意义就是，测量全网难度进而维持平均每 10 分钟产生一个新的区块的速度。挖矿难度改变的周期是两周，并没有什么特别的意义，只不过是一个权衡之下的结果。如果这个周期太短，难度会随着每一个周期找到的区块的数目的不同而波动（概率问题）。如果太长，整个网络的哈希算力会与难度大大地失去平衡（难度的调整滞后于计算能力的变化）。

每个比特币矿工独立地计算难度，只接受达到这个难度的区块。两个在不同分叉上的矿工可能会有不同的计算难度，但在同一个区块工作的矿工一定会对计算难度达成共识。

图 5.3 中显示，挖矿难度会随着时间不断地增加。其增加不一定是稳定线性或者是指数型，而是取决于市场行为。挖矿难度会受到有多少新矿工加入的影响，新矿工的加入本身又由比特币的当前价格来决定。总的来说，当越来越多的矿工加入并且挖矿的硬件设备效率越来越高，找到有效区块所花费的时间就会越短，紧接着难度就会增加，直到重新回到每 10 分钟找到下一个有效区块。

在图 5.3 中，虽然整个网络的哈希速度是平滑向上增长的，那条实线代表的难度却呈现阶梯函数式增长。这是因为每产生 2 016 个区块才会调整计算难度。

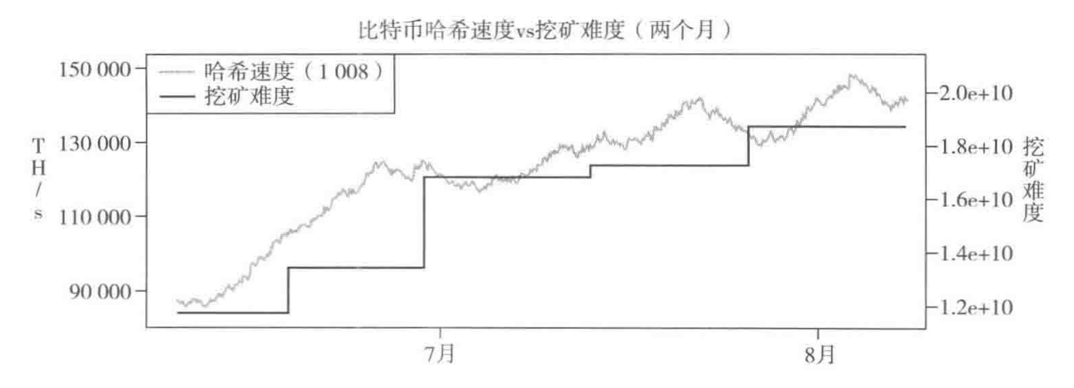


图 5.3　挖矿难度随时间变化（2014 年年中）

注：y 轴开始于 80 000TH/s（全网算力）。哈希速度是基于超过 1 008 个区块计算出来的平均值。
资料来源：bitcoinwisdom. com

可以从另外一个角度来看网络增长率，即看平均要多长时间才能找到一个有效区块。图5.4（a）展示出区块链上两个连续区块产生的间隔时间是多少秒。它逐渐下降，跳升，又逐渐下降。当然产生这种现象的原因就是每2 016个区块之后，难度重新被设定，找到区块的时间又重新回到大约10分钟。虽然一个调整周期内难度都不会变，但是随着越来越多的矿工加入，全网哈希算力增加而

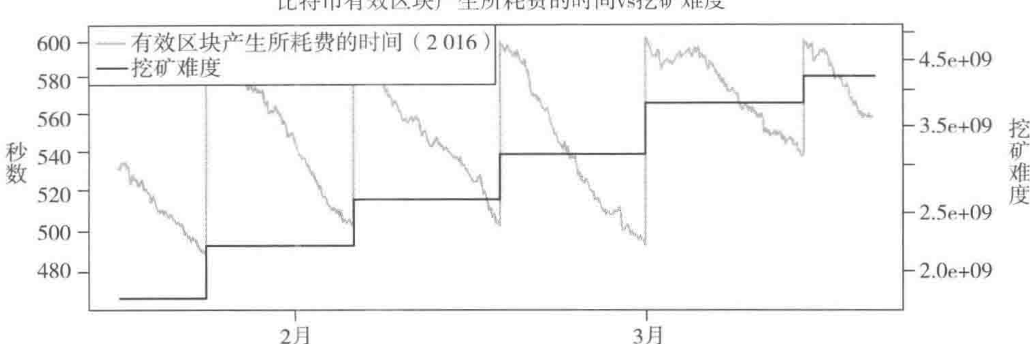


图5.4（a） 找到一个有效区块所花费的时间（2014年早期）

注：y轴开始于460秒。找到一个有效区块花费的时间是2 016个区块样本花费时间的平均值。由于当时全网挖矿速度的连续快速增长，找到一个有效区块所花费的时间在两周的时间内稳步减少。

资料来源：bitcoinwisdom. com

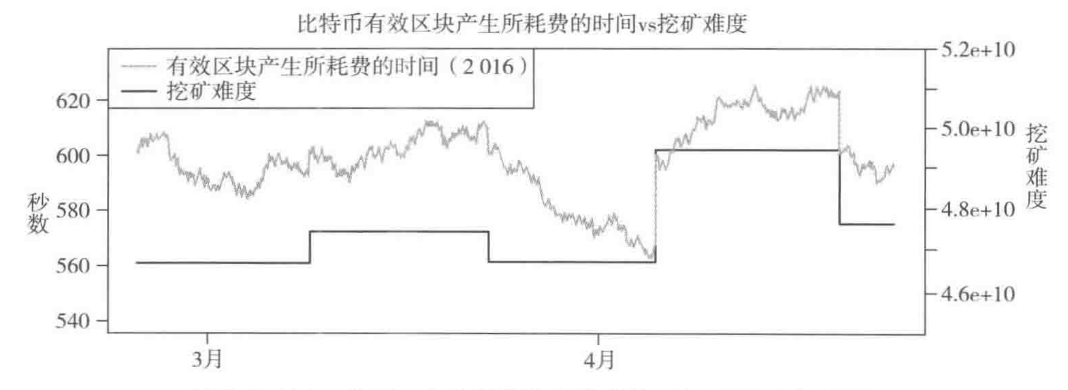


图5.4（b） 找到一个有效区块所花费的时间（2015年早期）

注：y轴开始于540秒。找到一个有效区块花费的时间是2 016个区块样本建造时间的平均值。由于整个网络增长开始减速，所以找到有效区块的平均时间非常接近10分钟，偶尔还会因为全网算力缩小而超过10分钟。

资料来源：bitcoinwisdom. com

难度不变，找到有效区块的速度越来越快，直到大概两个星期内2 016个区块被发现之后，难度会被重新调整。

即使找到有效区块的时间目标被设定为平均10分钟，但是在2013年和2014年的大多数时间，这个时间平均是9分钟左右，并且在两周的周期末尾阶段时接近8分钟。计算表明，全网运算能力的增速大概是每两周25%，或者每年几百倍，这个数字非常惊人。

不出所料，这种增长不会无休止地进行，2015年的增速就降低很多（偶尔也会负增长）。在图5.4（b）中，我们可以看到全网哈希算力达到了一个稳定的状态，发现每个区块的时间非常接近10分钟，甚至有时会超过10分钟，在这种情况下难度将会被调低。这种曾经被认为不可想象的状况却在2015年频繁发生。

虽然到目前为止，全网哈希算力还没有经历过灾难性的倒退，但是并不排除它发生的可能性。一个有关比特币崩盘的设想被称作“死亡螺旋”，不断下降的比特币价格导致挖矿无利可图，矿工们因此而退出挖矿（全网运算能力下降），继而进一步导致价格下跌。

5.2 挖矿所需硬件

前面我们谈到矿工所要做的计算是十分困难的。我们现在谈一下为什么计算如此困难，以及看一看矿工用来进行这些计算所用到的硬件设备。

矿工计算难度的核心在于，对SHA－256哈希函数的运算。我们在第1章抽象地讨论过哈希函数。SHA－256是一个通用的密码学哈希函数，它是在2001年被标准化的密码学哈希函数大家族里的一员。SHA是安全哈希算法（Secure Hash Algorithm）的简称。SHA－256是一个不错的选择，因为它是比特币被发明时可用的密码学哈希函数中保密性最强的。虽然它的安全性有可能随着时间推移而慢慢降低，但至少现在它还是很安全的。SHA－256的设计来自美国国家安全局（NSA），这也导致了一些阴谋论的诞生，但是并不影响它是一个很强的

哈希函数的事实。

近距离了解 SHA-256

图 5.5 展示了 SHA-256 运算的具体细节，虽然我们不需要知道比特币工作原理的所有细节，但是对矿工计算任务的大概了解是很有帮助的。

SHA 家族

SHA-256 名称中的“256”代表它有 256 位的状态和输出，技术上来说，SHA-256 是 SHA-2 函数家族中几个密切相关的函数成员之一，包括 SHA-512（它有更大的状态位，所以也更加安全）。还有一个是 SHA-1，这是一个有 160 位输出的早期函数，虽然目前认为安全性不高，但是同样应该在比特币脚本里。

整个 SHA-2 家族，包括 SHA-256 在密码学上的安全性是得到公认的，而下一代产品 SHA-3 家族已经从一个公开的竞赛（由美国国家标准与技术研究所举办）中诞生了。SHA-3 目前正在进行最后阶段的标准化测试，但在比特币出现的时候，它还没有出世。

SHA-256 是一个 256 位的状态机。这 256 个状态被分割成 8 个 32 位的字段，这样它可以最优化地运行在 32 位的硬件上。每一轮运算选择一定数量的字段——有些会进行一些小的逐位调整——最终进行 32 位模加法运算（modular addition），然后运算结果被移到状态最左的第一个字段，这样使得整个状态进行向右位移。这种设计的思路来自简单位的线性反馈移位寄存器（Linear Feedback Shift Registers，简称 LFSR）。

图 5.5 展示了一轮 SHA-256 的压缩函数运算，一个完整的 SHA-256 运算要做 64 次这样的迭代运算，在每一轮运算中，会使用稍微不同的常数，所以所有的迭代运算都是不一样的。

矿工的任务就是尽可能快地进行这种函数运算。矿工们互相比拼运算速度，

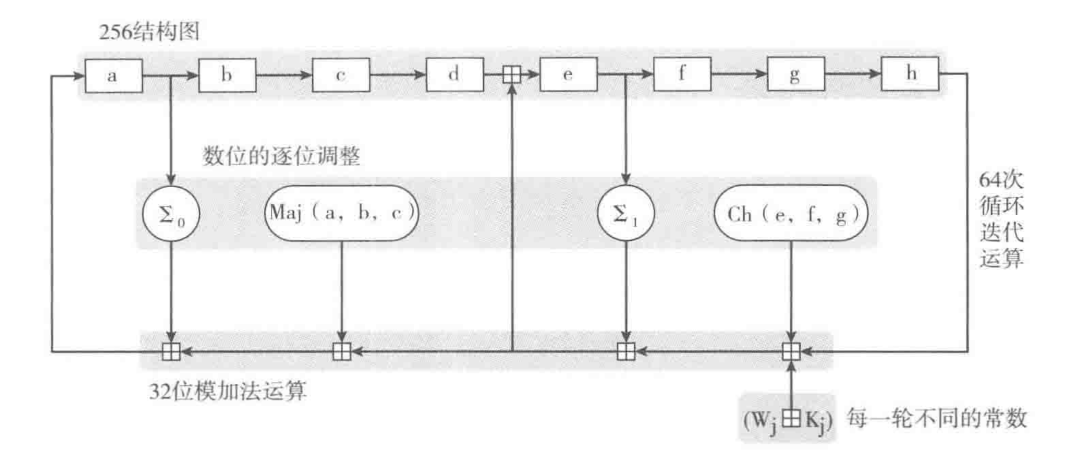


图 5.5　SHA－256 的结构

注：这是一轮压缩运算。Maj 是按位运算的。Ch 也是按位运算的，根据第一个输入值的不同而决定是选择第二个还是第三个。Σ_0 和 Σ_1 通过按位循环和⊕运算来操纵 32 位的字符输入。

算得越快收益越高。为了实现尽可能快的速度，矿工需要进行 32 位字段操控，32 位模加法运算，同时做按位逻辑运算。

我们很快就会看到，在比特币机制下，为了得到供其他节点使用的哈希函数，实际上要求两轮 SHA－256 运算。这是比特币的奇怪之处，进行两次运算的原因并不清楚，但这就是比特币的个性，作为比特币矿工只能服从。

CPU 挖矿

第一代挖矿工作都是在普通电脑上完成的，也就是用通用中央处理器（CPU）来进行运算。事实上，CPU 挖矿的工作就像图 5.6 中代码所示的逻辑那样简单，也就是说，矿工简单地按照线性的方式尝试所有的临时随机数，在软件中进行 SHA－256 的运算，并检查结果确认是否找到一个有效区块。请注意，正如我们之前提到过的，这段代码要进行两次 SHA－256 运算。

普通电脑运行这段代码到底有多快？一台高端的个人桌面电脑，每秒可以计算大约 2 千万次哈希函数（20MH/s），按照这个速度，根据 2015 年早期的难度水平（2^{67}），大概需要几十万年来找到一个有效区块。毫不夸张地说，挖矿

```
TARGET = (65535 << 208) / DIFFICULTY;
coinbase_nonce = 0;
while (1) {
     header = makeBlockHeader(transactions, coinbase_nonce);
     for (header_nonce = 0; header_nonce < (1 << 32); header_nonce++){
          if (SHA256(SHA256(makeBlock(header, header_nonce))) < TARGET)
               break; //block found!
     }
     coinbase_nonce++;
}
```

图 5.6　CPU 挖矿的伪代码

真的非常困难！

如今使用一个普通电脑用 CPU 挖比特币，在目前的难度下已经无利可图了。在过去的几年里，用 CPU 挖矿的旷工可能会非常失望地发现，他们永远不可能通过挖矿赚到钱，因为他们不了解比特币是如何运行的。

GPU 挖矿

第二代矿工意识到用 CPU 挖矿是在做无用功，他们开始用显卡或者图形处理器（GPU）来挖。

几乎每一个现代个人电脑都有内置的 GPU 以支持高性能图像处理，这些 GPU 都有高吞吐量和高并行处理功能，这两点对比特币挖矿都非常有利，比特币挖矿存在大量的并行处理，因为你需要同时用不同的临时随机数计算多个哈希值。2010 年，有一门计算机语言开放运算语言（Open Computing Language，简称 OpenCL）诞生了，这是一个可以使 GPU 进行非图像处理类工作的通用语言。OpenCL 是一门高级语言，人们可以用它在显卡上做很多种类型的运算，而且速度比在 CPU 上的要快。这给通过 GPU 来进行比特币挖矿铺平了道路。

当时，通过显卡来挖矿有好几个吸引人的地方。首先，买显卡很容易，而且哪怕是业余爱好者也能轻松配置显卡。你可以在网上或大多数专营电子产品的商场里买到它。对大众来说，显卡是最容易获得的高端硬件设备。其次，显卡还有一些格外适合比特币挖矿的特性：显卡的并行性设计使其具备很多算术逻辑单元（Arithmetic Logic Units，简称 ALU），可以同时进行 SHA－256 运算。

有一些 GPU 还特别集成了针对位移操作的指令，这对 SHA-256 的运算非常有用。

大多数显卡都可以超频，这意味着如果你愿意承担显卡过热或者出现故障的风险的话，你可以让显卡以高于设计频率的频率更快地运行。超频是游戏玩家们渴望了多年的特性。对于比特币挖矿来说，超频会增加收益，即使超频可能引起一些运算错误。

举例来说，将显卡超频 50%，也就是说运算速度加快 50%，同时可能会造成 SHA-256 运算出错概率增加 30%。如果显卡错误地接受了一个不正确的运算结果——虽然不太可能发生——你还可以通过 CPU 来进行二次确认。然而，你可能永远都不会知道一个正确的运算结果被错过了。但是通过超频产生的运算速度的增加，完全可以抵消由于显卡运行错误产生的正确输出减少，这样超频还是合算的（从经济效益上来说）。在上述的例子中，超频使得吞吐量增加到原来的 1.5 倍，而运算成功率降低到了 0.7，那么乘积就是 1.05，也就是意味着超频使得你的获利增加了 5%。为了最大化收益，人们花了很多时间去优化最佳的超频比例。

最后，你可以通过一个 CPU 和一个主板加载许多个 GPU。然后你便可以在安装了多个 GPU 的电脑上运行比特币节点，监听网络收集交易，组装区块，同时用多个 GPU 进行 SHA-256 的运算，以更快地找到正确的临时随机数及其对应的有效区块。很多人创造性地发明了很多有趣的“自制型”硬件设置，如图 5.7 所示，用一个 CPU 来驱动很多个 GPU。这种情况仅发生在比特币的早期，当时大多数矿工都是比特币的爱好者，他们并不具备服务器搭建及运营经验。但是他们还是做了很多巧妙的设计，使得大量的 GPU 可以在一个较小的空间里同时运行，同时还解决了散热的问题。

GPU 挖矿的缺点

GPU 挖矿也有缺点。GPU 有大量的内置硬件来进行图形处理，这些特定硬件对比特币挖矿没有任何用处，比如它们大量的浮点运算单元（floating point units），在 SHA-256 的运算中完全用不到。

矿工和游戏玩家的对比

根据民间传说，2011 年，由于比特币矿工采购了太多的显卡以至于影响到了正常的市场需求，这造成了比特币社区和游戏社区之间的摩擦，因为游戏玩家们发现采购某个热门显卡变得越来越难。有趣的是，尽管如此，很多失望的游戏玩家因此而对比特币产生了兴趣，甚至有些游戏玩家因此而变成了比特币矿工！



图 5.7 一个用于比特币挖矿的家庭组装式 GPU 机架

资料来源：István Finta，bitcointalk. org

同时 GPU 也没有很好的冷却处理设置，尤其是当你把大量的 GPU 堆放在一起的时候，这个问题就尤为突出。设计显卡的时候并没有考虑如图 5.7 所示的这种堆放的情形，原始的设计场景就是在一台电脑、一个机箱、一块显卡运行做图形处理而已。

GPU 也非常耗电，所以一台普通的电脑也会消耗很多电。由此引发的另一

个缺点就是，你要么自己构建特定的主板，要么花大价钱购买可以搭载大量显卡的特定的主板。

一个非常高端的显卡经过超频之后可能使得运行速度达到200MH/s，也就是说，每秒可以进行2亿次哈希运算，这是用CPU不可能达到的一个数量级。但是即便如此，即使你将100块这样的显卡集成在一起进行运算，根据2015年早期的比特币挖矿难度，仍旧需要运算几百年才有可能找到一个有效区块。因此，用GPU来挖矿基本上已经成为历史，但是在其他一些另类币的早期阶段还是很有效率的。

现场可编程门阵列挖矿

2011年左右，用于现场可编程门阵列（Field-Programmable Gate Array，简称FPGA）的硬件设计语言Verilog，第一次用于比特币挖矿。一些矿工开始用FP-GA来代替GPU进行挖矿。

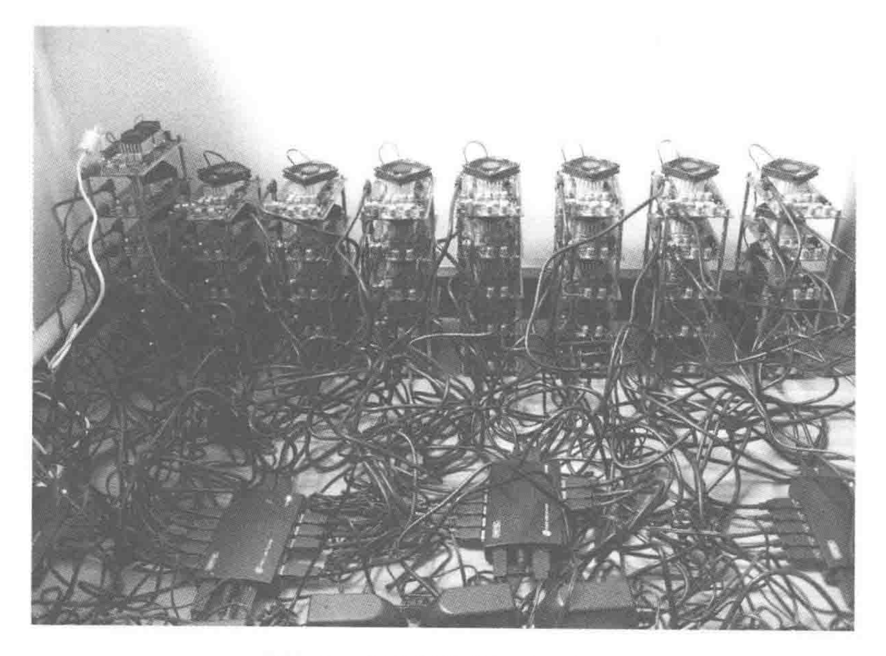


图5.8　家庭组装式FPGA机架

资料来源：Xiangfu Liu，www.openmobilefree.net

FPGA 的工作原理是在追求定制硬件的最佳性能的同时，用户可以现场调试或者修改硬件参数。相比之下，常用的硬件是在出厂之前就设计好的，以后是无法更改定制而只能永远做同样的工作。

FPGA 比 GPU 的性能好，特别是在数位操作（bit fiddling）方面 FPGA 很轻易就可以做到。FPGA 也很容易冷却，不像 GPU，FPGA 理论上可以使用硬件板卡上的每一个晶体管进行挖矿运算。跟 GPU 一样，可以将很多 FPGA 堆叠在一起，通过一个中央处理单元来驱动所有的 FPGA，如图 5.8 所示。总体来说，相比显卡堆叠，我们可以构建一个更加干净整洁的大型 FPGA 阵列。

精心使用 FPGA 可以使得运算速度上升到 1GH/s，也就是每秒 10 亿次哈希运算。这显然比 CPU 或者 GPU 在性能上都有很大的提高，不过即使你有 100 块每秒运算 1GH/s 的 FPGA 板，在 2015 年早期的比特币难度之下，平均仍旧需要 100 年的时间才能找到一个有效区块。

虽然性能提高了，但由于以下几个原因，用 FPGA 挖矿的时代也非常短暂：首先，使用 FPGA 来挖矿其实更加困难——几乎需要一直超频使用——这远超 FPGA 供普通消费者而设定的频率。因为这个原因，很多人在用 FPGA 挖矿的时候经常看到各种报错和故障。其次，优化 FPGA 的 32 位加法处理上十分困难，而这在 SHA-256 的运算中非常关键。最后，FPGA 在多数商店都买不到，而且相比 GPU 来说，只有少数人才知道怎么搭建 FPGA 并进行相应的编程。

最重要的是，即使在性能功耗比方面，FPGA 相比 GPU 的提升也不是很高，这就使得用 FPGA 挖矿只是一个短期现象。尽管 GPU 挖矿的时代持续了大约一年，而 FPGA 存在的时间更加有限——仅仅存在了几个月，之后定制化的专用集成电路技术（Application Specific Integrated Circuits，简称 ASIC）就诞生了。

专用集成电路技术挖矿

当今的挖矿市场主要被 ASIC 所主导。这些 IC 芯片（集成电路芯片）被设计、制造、优化，就是为了比特币挖矿这个唯一目的。有几个大型的供应商出售这些芯片，消费者可以买到不同种类的 ASIC 矿机——较大型但略微昂贵点的款式，或者更加小巧的，当然还有一些可以节省能源的环保型等。

设计 ASIC 芯片需要非常专业的知识，它们所需要的研发周期也比较长。尽管如此，比特币 ASIC 芯片的设计制造过程（从发现问题到制作出解决问题的芯片）出乎意料地迅速，甚至打破了芯片行业的业界纪录。但弊端是前几代的早期芯片的设计有许多缺陷，而且大多数芯片没有达到它们所承诺的性能指标。但现在的比特币矿机芯片技术，已经相当成熟可靠了。

到 2014 年年底，ASIC 芯片的寿命十分短暂，原因是整个网络的运算能力不断地快速上升。绝大部分早期的 ASIC 芯片在 6 个月后就被淘汰了。在这段时间里，大部分的利润都是在早期实现的（后期几乎没有什么利润了）。矿工往往在 ASIC 芯片“保鲜期”的前 6 个星期可以实现整个利润的一半，这就使得芯片的出货速度显得至关重要。基于这个行业的不成熟，许多矿工经常遇到延迟出货的情况，有些芯片送到客户手上的时候已经快要被淘汰了。由于比特币的全网运算能力已经稳定下来，现在的比特币挖矿设备有比较长的寿命，但在早期，失望的客户对供应商的欺诈控诉的事件时有发生。

在比特币历史的大部分时间里，挖矿的经济效益对小矿工来说一直不是很好，这些小矿工们需要通过在线预定矿机，等待矿机生产送货，然后再开始挖矿赚钱。实际上，大多数情况下，由于不断增加的挖掘难度，很多人是一开始就注定要亏钱的。好在到 2013 年的时候，比特币价格大涨，彻底扭转了比特币矿工亏钱的状况。实际上，挖掘比特币一直是一个很昂贵的投资，因为要赌注比特币的价格会上升，即使是在挖掘比特币中赚钱的那些矿工，如果能够把投资于挖掘设备的钱直接用于购买比特币，并在盈利时卖掉，这样他们的状况会更好些。

如今你仍然可以购买矿机，我们也不会劝阻你通过这种方式去了解比特币和加密数字货币，但是我们再次强调，这不是一个明智的生财之道。考虑到矿机运行所需要耗费的电力成本以及冷却成本，大多数 ASIC 矿机都无法靠挖矿来赚回成本。

如今：专业挖矿的天下

在今天，挖矿已经从个人领域转到了大型专业挖矿中心。为了保持竞争优

势，运作这些挖矿中心的公司不愿意公布其运营细节。据猜测，这些运营者大量采购打过折的更新的功效更高的 ASIC 矿机而不是采购那些能够直接出售给个人的 ASIC 矿机来维持利润。

图 5.9 就是一个在格鲁吉亚运作的挖矿中心的图片。



图 5.9　Bitfury 的挖矿中心

注：在格鲁吉亚运作的专业挖矿中心。

要建立一个挖矿中心，需要具备三个重要因素：气候、电费、网络接入速度。尤其是要找一个气温偏寒的地方，这样可以节省冷却费用（大型计算中心会产生大量的热量，如不马上降温，会损害矿机）。冷却是比特币挖矿最大的挑战，其挖矿本身的耗电量，用单位面积来算要超过传统的数据中心，所以也会产生差不多当量的热量需要冷却。当然那里的电费要便宜。另外，接入网速要快，使得与比特币网络中的其他节点更快速地链接，这样当新的区块被广播出来的时候，你可以很快监听到。基于这些考量，所以格鲁吉亚和冰岛成为比特币挖矿中心的首选之地。

与挖金矿的相似之处

尽管比特币挖矿这个名词起得很有意思，如果回顾比特币挖矿的发展历程，

我们就不难发现其与历史上的挖金矿有着有趣的相似之处。它们都开始于类似的淘金者热潮，很多年轻人和业余爱好者积极地参与其中。

比特币挖矿经历了一个逐渐演化的过程：从 CPU 到 GPU，再到 FPGA，最终达到现在的 ASIC。而历史上的挖金矿则是从个人拿着盘子在沙里淘金，到一小群人用流沙槽来淘金，再到一群人用水冲刷金山来淘金，直到现代机械化露天挖矿（如图 5.10）。比特币与黄金都从个人操作为主逐步演变为大公司专业运作。另外一个相似点就是，大多数的利润都被设备制造商拿走了，不管是黄金采掘设备还是比特币 ASIC 矿机生产商，而埋单的都是那些希望一夜致富的人。

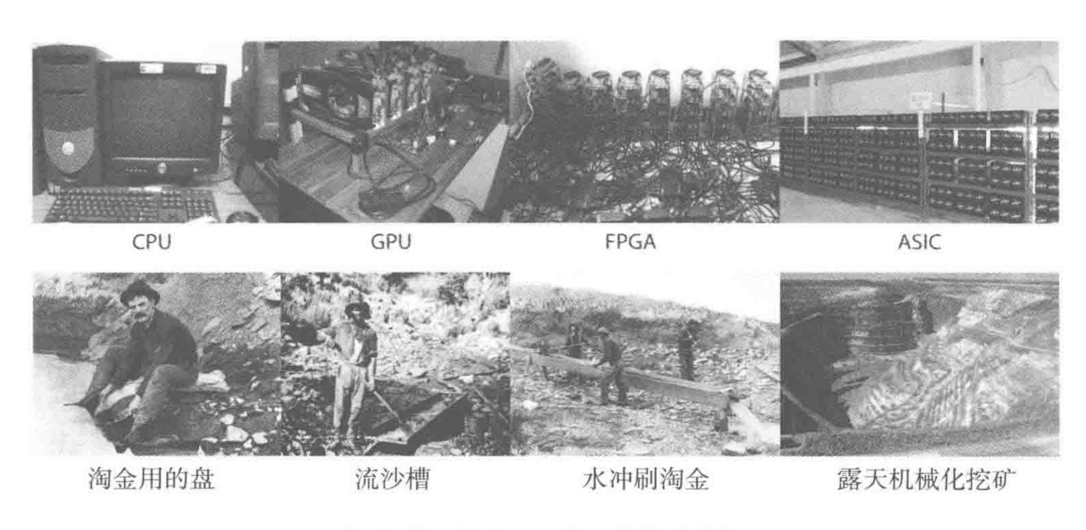


图 5.10　比特币和黄金的挖矿进程

注：我们可以看到，比特币挖矿和黄金采掘进程有一个清晰的类似进程，两种活动在最初都对个人用户很友好，但是随着时间的推移，被大型公司采取集中式大批量运作控制。

未来

现在，用 ASIC 挖矿是唯一一种可以赚钱的比特币挖矿手段，但对个体矿工来说，是十分不友好的。人们不禁要问，未来会如何发展？小规模矿工是否永远不可能再参与到比特币挖矿中？是否有办法把小规模矿工重新纳入挖矿体系中去？更重要的是，现在使用的 ASIC 和专业挖矿中心是否已经违反了比特币当

初设计的初衷：一个完全去中心化的系统，在这个系统上里每个人都能用自己的电脑去挖矿。

此外，如果这已经违背了中本聪对比特币的最初设计，换成系统只允许CPU来挖矿是不是更好？我们将在第8章探讨这些问题，以及一些对ASIC不友好的替代方案。

自我循环周期

事实上，有一些规模较小的另类币已经使用了和SHA-256不同的解谜算法，但是它们的挖矿发展轨迹和比特币没有什么不同。我们将在第8章到第10章更深入地讨论这些另类币，但是请记住，ASIC的研发和生产有着比较长的时间周期，所以如果一个使用新的解谜算法的另类币（即使只是在SHA-256的基础上做一点修改），在有针对性的ASIC面世之前还是会有一段时间。通常跟比特币一样，其他另类币的挖矿发展也会经历从CPU到GPU，再到FPGA，或者直接到ASIC的过程（前提是这个另类币非常成功，比如莱特币）。

因此，小规模矿工的策略也许应该是尝试一些新的另类币，在它们的价值还没有足够大到吸引大型挖矿集团投资的时候，成为这些另类币的挖矿先行者，就跟黄金采掘的过程一样，小规模矿工可以去尝试那些还没有被证明储量的区域。当然，这也意味着先行者们将会面临一个重大的风险，也就是这些另类币有可能永远不会成功。

5.3 能源消耗和生态环保

我们看到大型职业化挖掘中心是如何接管了比特币的挖掘工作，我们也看到比特币挖掘与历史上的淘金热有多么类似。时至今日，金矿开采一直被环保问题所困扰，比特币挖矿虽然还没有达到那个程度，但它已经开始消耗大量能源，这已经成为热门话题。本节中，我们将着重讨论比特币挖矿的能源消耗问题，以及其对货币系统和地球生态的影响。

热力学限制

根据热力学里的蓝道尔原理（Landauer's principle，蓝道尔是前苏联20世纪60年代天才物理学家），任何一个不可逆转的计算都会消耗一定的能源，逻辑上来说，这种计算也可以被认为是一种信息丢失的过程。蓝道尔原理特别指出，任何移位运算都会消耗一定量（k Tln 2）的焦耳，其中k代表玻尔兹曼常数（Boltzmann constant，大概等于 1.38×10^{-23} J/K），T代表芯片以开尔文为单位的温度，ln 2代表2的自然对数，大约等于0.69。算下来每一个单位数据的运算会消耗一点点热量，这从基础物理学原理上提供了一个能源最低消耗下限。

这里我们不做进一步推导，大概的意思就是每进行一个不可逆的数位运算都会消耗一个最小量的焦耳，能源是永远不会被摧毁的，只会从一种形式转变成另外一种形式，在计算中所消耗的能源大多数都是从高等级的电能转换过来的，然后被转换成可以在环境中最终消失的热能。

作为一种密码学中的哈希函数，SHA-256就是一个不可逆的运算，我们可以回忆一下第1章里所说的，不可逆转是作为密码学哈希函数的一个基本要求，既然不可逆运算需要消耗能源，那么SHA-256作为比特币挖矿的基本要素也是不可逆的，那么比特币的挖矿过程必定会消耗能源。蓝道尔原理中描述的能源消耗下限要远低于实际挖矿过程所消耗的电能，虽然我们目前无法使计算的能源消耗达到这个热力学原理中的最优消耗，但即使我们做到了，比特币挖矿也是要消耗能源的。

比特币挖矿是如何消耗能源的？这个消耗过程分三个部分，其中有些可能还不是很明显：

1. **内涵能源**。首先，比特币挖掘设备需要被生产出来，生产时所用的原材料就需要被物理开采出来，然后要把这些材料通过一系列的生产流程转化为比特币挖矿专用的ASIC，这两个过程都需要消耗能源，这被称为内涵能源。在收到那些矿机的时候，你已经消耗了巨大的能源——当然包括物流过程中产生的能耗——即使这时候你还没有开启这些矿机！

可喜的是，随着越来越少的挖掘容量的出现，内涵能源的消耗就会降低。

随着越来越少的人会去购买新的 ASIC 矿机，这些矿机被淘汰的速度也会减慢，那么相应的内涵能源也会在多年的挖矿中被摊销。

2. **电能**。当矿机启动开始挖矿时，它就会消耗电能。根据蓝道尔原理，这一步肯定会消耗能源。随着矿机越来越高效，所消耗的电能也随之下降，但是根据蓝道尔原理这个消耗不能降为 0，电能消耗将会伴随着矿工的挖矿生涯。

3. **冷却**。比特币矿机需要被冷却，这是为了防止矿机出故障。如果在非常寒冷的环境中进行小规模的挖矿运营，冷却成本会微不足道。但即使是在非常寒冷的环境中，一旦在一个很小的空间运行了足够多的 ASIC，还是需要承担额外的冷却成本去解决散热问题。通常冷却挖矿机的耗能形式也是利用电力。

大规模挖矿

内涵能源和电能的消耗（每单位挖矿工作完成）会随着挖矿运营规模的增加而降低，设计和制造运行在大型数据中心的芯片本身单位成本会降低，同时由于不需要很多电源，你可以使得电力输送更加有效。

当讨论冷却问题的时候却恰恰相反，冷却成本会随着规模的增大而上升。如果要进行一个大规模的比特币挖矿运营，需要在一个地方运行大量的矿机，那就意味着空间比较小不易于散热。冷却成本会随着规模化而增加（每单位运算量），除非矿机运行的物理空间同等规模地增加。

能耗预估

整个比特币系统到底需要耗费多少能源？当然，我们无法做到精确统计，因为这是一个去中心化的网络，大量的矿机分散在各处，并且没有正式记录。但是有两种基本方法可以对比特币矿机所产生的能耗进行估算。根据 2015 年早期的比特币价格，我们可以进行一个快速的简单计算，我们必须强调一下，这个数字只是一个大概的估算，因为不管哪种方法，计算过程中所用的参数都是很难估计并且变化很快，这些结果只能是一个数量级上的估算。

自上而下

第一种是自上而下的方法。现在每一个区块奖励是 25 个比特币，大约值 6 500美元。也就是说，比特币体系平均每秒钟凭空产生 11 美元给矿工。

现在我们思考一个问题：如果矿工把所有的 11 美元都用在电费上，他们可以买到多少电？当然，矿工通常并不会把全部的收入都用于电费，这是用于计算电费的上限。电价在各地的差异非常大，我们可以用美国的工业电价，大约每千瓦时（kwh）10 美分的价格来预估，也就是每百万焦耳（megajoules，简称 MJ）大概 3 美分。如果比特币矿工把所有的每秒 11 美元收入都用来支付电费，他们可以购买每秒 367 百万焦耳，消耗大概 367 000 千瓦时电力。

> **单位能耗和单位电力** 国际单位制（SI）中，能耗的衡量单位是焦耳，电力的衡量单位是瓦特，1 瓦特代表每秒钟 1 焦耳。

自下而上

第二种是自下而上的方法，通过观测每个区块的难度，了解矿工计算的哈希数量，并以此来进行估计。假设所有的矿工都使用最高效的矿机，我们可以推导出一个最低电耗。

目前，最好的商业化矿机的功效数值差不多是 3GH/s/W[①]。那就是，这样的 ASIC 矿机每消耗 1 瓦特的电力，可以进行每秒 30 亿次哈希函数运算。目前全网算力是 350PH/s，也就是 350 000 000GH/s。[②] 根据这两个参数计算，我们就可以知道目前基于这种矿机效率，每秒钟全网的矿机需要消耗 117MW 的电力。当然这个数值还没有包括所有冷却需要消耗的能耗以及芯片本身的内涵能耗。因为只是做一个能耗的下限估计，这么算是可行的。

结合上述两种方法，可以推导出比特币挖矿大概所耗电力，这是几百万瓦特（megawatt，简称 MW）的数量级。

---

① GH 为 gigahash，s 为 second，w 为 watt。——译者注

② 截至本书翻译的时间，全网算力已经增长到了 1 200PH/s。——译者注

100万瓦特究竟是多少？为了便于直观理解，可以对比一下大型发电厂产生多少电力。世界上最大的发电厂之一，中国的三峡水电站的发电总量是10 000MW，一个普通的大型水力发电厂的发电总量一般是1 000MW。世界上最大的核电站日本柏崎刹羽核电站（Kashiwazaki-Kariwa）的发电总量是7 000MW，而平均来说核电站的发电量为4 000MW，而火电电厂的发电总量一般为2 000MW。

根据我们的估算，整个比特币网络大概消耗了一个大型电厂总发电量的10%。虽然这个数字已经相当惊人，但是和地球上其他的用电“大户”比起来，这个还算是小的。

比特币挖矿在浪费能源吗

比特币这种“浪费”能源的形式经常被人诟病，因为SHA－256的运算没有其他任何用处。但是我们必须认识到任何一种支付系统都需要能源和电力的消耗。就拿传统的货币来说，纸币印刷、ATM机器的运行、硬币分类机器、点钞机、支付服务系统以及运送现钞和金条的武装押运车，无一不在消耗各种能源。你也可以一样说这些能源的消耗除了维护整个货币体系之外，也没有任何其他用处。所以，如果我们认可比特币作为一个有用的货币体系，那么支持比特币体系的能耗就不能认为是浪费。

当然，如果我们可以用更加节省能源的解谜算法来代替现在的比特币挖矿，同时确保货币的安全性，那自然更好。我们将在第8章讨论这个问题，然而我们并不知道这种可能性是否存在。

能源的循环使用

另一种使比特币更加环保的主意是，把挖矿过程中产生的热能进行二次利用，而不是让热能无谓地耗散在空气中。这种收集计算机运算所产生的热能的模式被称为“数据火炉”（data furnace）。这个想法的原理是使比特币矿机挖矿产生的热能经过一种特殊供暖装置的转换，用来进行家庭供热，而不需要传统的电取暖器。这部分热能供给就成了比特币挖矿的副产品。这么做的效率其实并不比购买一个传统的电取暖器差。也许对于家庭消费者来说，使用一个“数

据火炉”并不会比将供暖设备连上网络和电源插座更复杂。

这种方案也有一些问题。虽然矿机发热的效率和电取暖器差不多，但是它们本身比用天然气供暖的效率差很多。另外如果在夏天每个人都把矿机关闭（至少在北半球），那么比特币的全网算力将会伴随人类取暖需求而产生季节性变动。如果数据火炉方案真的推广开来，将会给比特币的共识机制带来很有有趣的影响。

矿机的所有权也不明确。如果买了比特币数据火炉用于取暖，你是否拥有挖矿所获得的收入呢？还是出售设备的公司获取这部分收入？大多数人对比特币挖矿完全不感兴趣——有可能永远没兴趣——所以由出售这些设备的公司来获取这部分挖矿收入更合理。这也就意味着取暖器会以略微亏损的价格出售。这样一来，一些有创造性的用户可能会在购买了这些取暖器之后，对设备进行改造以使得他们自己可以获取这部分挖矿收入。这可能会引发令人难堪的数据所有权管理之争。

将电力转换成现金

长远来看，比特币产生的另一个问题是：它可以最有效率地把电力转换成现金。想象一下，如果比特币 ASIC 矿机是一个很容易购买到的商品，并且主要的挖矿成本是电力，这便意味着，提供免费的或低成本的电力将会面临被滥用的风险。

在很多国家，政府都有用电补贴，特别是对工业用电进行补贴。这么做的原因是政府希望吸引工业投资留在本国。但是由于比特币提供了一种很好地把电力转换为现金的途径，这可能使得政府要重新考虑用电补贴的模式，以防它们补贴的电力全部被转换成了比特币。政府用电补贴的意图是，希望可以吸引那些对国家经济和人民就业有帮助的企业，但是补贴比特币挖矿也许并不能对这两点有所帮助。

更大的问题是全球有数以亿计的“免费”插座，分布在家、学校、酒店、机场以及办公大楼等。有人可能把挖矿设备接在这些地方挖矿，因为别人会为此支付电费。事实上，他们还可能会用过时的设备而压根不考虑升级，反正电

费又不是他们支付。在全世界范围内监控这些用于比特币挖矿的“偷电”行为，是一个异常艰巨的任务。

5.4 矿池

设想一下作为单个矿工。假设你花了辛苦赚来的6 000美元买了一台全新闪亮的比特币矿机，你所期望的性能是平均每14个月会找到一个有效区块（在2015年早期一个区块的奖励价值在10 000美元）。

考虑到电费和其他运营成本，矿机的平均收入期望值应该是每个月400美元。如果可以确定每个月都能获得400美元，那么购买一台矿机是合理的投资。但是别忘了，挖矿是一个随机过程，你不知道什么时候可以发现下一个有效区块。在找到有效区块之前，什么都赚不到。

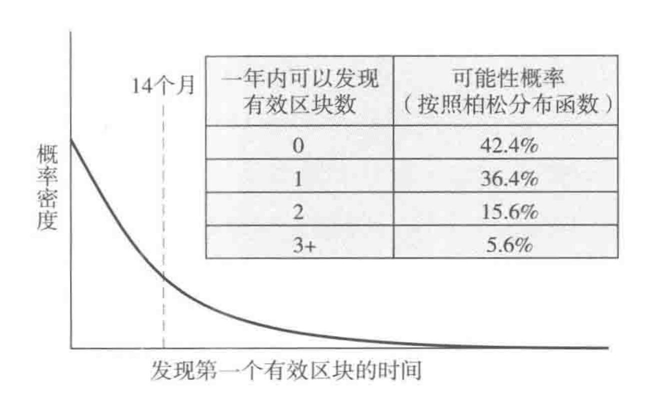
高方差

从矿工第一年能找到有效区块数的概率分布上看，这个分布差异是很大的，期望值（也就是第一年能找到区块的平均数）是相当的低。因为发现区块的比率是一个很低的固定值，并且这个值和你上次发现一个有效区块所花费的时间完全没有关系，因而总的发现区块的期望值是以柏松概率分布[①]。柏松分布是指，如果有N个独立事件，每个事件成功的概率是λ/N，当N接近于无限大的时候的成功概率分布。比特币挖矿中，尝试每一个临时随机数的行为实际上就是一种超小成功概率事件，所以即使对于小矿工来说，N的值也确实很大，这种近似类比是很合适的。

如果你期望每14个月找到一个有效区块（根据泊松分布可知λ = 6/7个有效区块/每年），则有超过40%的概率在第一年你不会找到任何有效区块。对于

① 柏松是18世纪法国数学家，概率学奠基人之一，柏松分布被广泛用于各个领域的概率分析。——译者注

个体矿工来说，这可能是灾难性的。你在一个矿机上花费了数千美元，并且支付了很多电费来运行，结果什么都没有获得。第一年能获取一个有效区块奖励的概率大概是36%，这也就意味着即使你的电费不高，你也就可能刚刚够支付电费。当然也有很小的概率可能会发现两个甚至更多的有效区块，这种情况下才有可能真的赚钱。详见图 5. 11。



一年内可以发现有效区块数	可能性概率（按照柏松分布函数）
0	42.4%
1	36.4%
2	15.6%
3+	5.6%

图 5. 11　挖矿成功的不确定性

注：假设全网哈希算力是不变的，平均发现一个区块的事件是 14 个月，对于一个小矿工来说这个成功概率的波动太大了。

这些数字只是一个近似估算，但主要的意思是，即使挖矿从期望值来说是合理的，也就是说，投资有足够的回报，但由于方差足够大以至于会有很大的概率什么都得不到。对于一个小矿工来说，这也就意味着挖矿就是一个赌博游戏。

矿池

历史上当小商人遇到大风险的时候，他们会自发组建一个互助保险公司来降低风险。比如，农夫会自发地聚在一起形成一个协议，如果任何一个个体农夫的谷仓不小心被烧掉了，那么其他的农夫可以把他们的利润拿来和这个不幸的农夫分享。那么对于比特币的小矿工是否也可以用类似的方式来降低风险呢？

矿池应运而生——矿池就是一个比特币矿工互相之间的保险。一组矿工可以形成一个矿池共同进行挖矿，并指定一个币基接受人。这个接受人就是矿池

管理员。所以不管是谁最终发现了一个有效区块，矿池管理员将会收到这个区块的奖励，继而根据每个参与者所贡献的工作量按比例分配给所有矿池的参与者。当然，矿池管理员可能从中分一部分来作为矿池管理服务的收入。

假定每个人都信任这个矿池管理员，这样的分配安排可以极大地降低矿工成功寻找有效区块的概率波动。但是矿池管理员如何知道矿池里每个成员实际上到底贡献了多少工作量呢？同时他又是如何去分发收入的呢？很显然，矿池管理员不希望是靠每个成员的申明，因为他们可能会虚报自己的工作量。

挖矿工分

对于这个问题，我们有一个简洁的解决办法。矿工可以通过输出**挖矿工分**（mining shares）来证明他的工作量，工分就是那些接近有效区块的区块。比如目标值是个前面67位是零的数字，输出的哈希值必须要低于这个目标才算有效。在寻找这个哈希值的过程中，矿工可能找到其他一些区块，它们的哈希也有许多零，但达不到67个。矿工可以用这些区块来证明他们确实在工作（见图5.12），一个合格的工分可能要求40~50个零，取决于矿工所加入的矿池的要求。

```
9D1842A2A98DEDE34E00F6B8406AED0CE11BDC906C6DB6E23BCD9DE35DC4C339
86006DC06851F801FF0322E4CB92959DB619F19A03415B0C8FE131968005B9DA
00000000004D4120FD53C6CE8F013367209E905F4AE4D7837FFCFAA22B95CEDF
60E9D45D86FA3AE285615A3972E9F85C68FEA07611830F49ED15EEE1460E83A4
00000000007EF3D0D4479C9FB96FF100601618AD56BD240EF762B1B6842D1CF5
44AEE951CD30363A0A750C81CDC4BC0D3427DACA1C878A489120EB92430866F7
D048C51CA7EA5E6B61F6B40E739F9F35E2C653A37BE7D3EA2474F5E7777C8790
0000000000000001EB96F35E74E9B0F84BC921D52EDC878A754658F23313E86
1289F7CFA4A86DEBB743D2B94AAD0A916A9282FDCA05B70E72C627FE5A592959
561BBB9E8AAC2B1DDE1E163DA1E4F05BC1A9B1E92B04DCE834A6EB827C5E2E5B
000000000082D602D87B67A42ED2BF763E92A76BE90F76A9CA71AB958EB7657A
FD639DC38BB5279885F0FC42E7FD92D37ABD7FEAFD828CEDA2731CD781DC77D7
```

图5.12　挖矿工分

注：矿工不断尝试去发现哈希值低于目标区域的有效区块。在这个过程中他们会发现一些区块的函数值比目标值少了几个0——但是已经是非常稀有的，这证明了他们确实在进行繁重的运算工作。在这个图中，浅色阴影的哈希值就代表工分，深色阴影部分是有效区块的哈希值。

矿池管理员也会作为参与者之一运行比特币节点，收集交易并组装区块。他会把他自己的接收地址放在币基交易里，然后把这个区块发给所有矿池里的矿工们。矿工们收到后会在这块上面开始挖矿，最后递交工分来证明他们确实进行了运算工作。

当矿池一个成员找到了一个有效区块，他会把这块发给矿池管理员，然后管理员会根据大家的工作量按比例分配奖励。发现这个有效区块的矿工并不会因此获得特别奖励，所以如果其他矿工的工作量更大，那么其他矿工就会获得更多的奖励，即使他们并不是真正发现有效区块的人。如图 5.13。

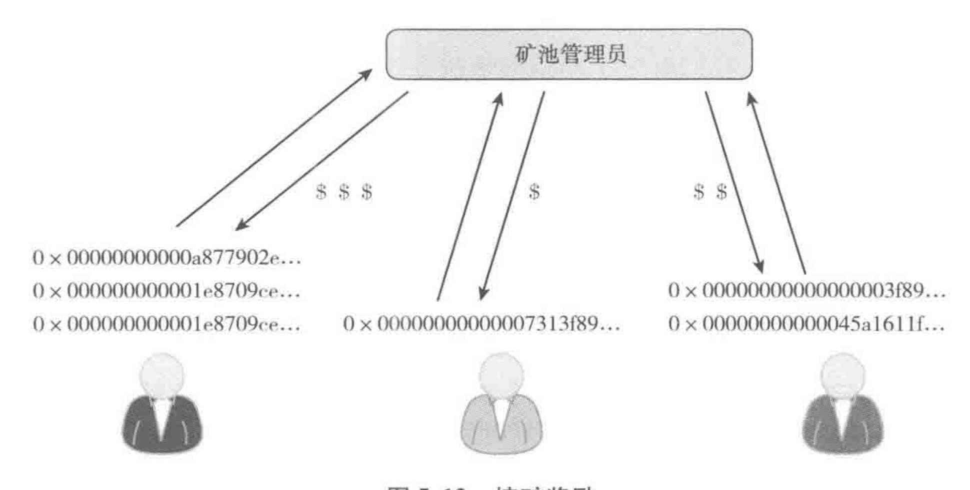


图 5.13　挖矿奖励

注：图上三个矿工在同一区块上挖矿。他们最后的奖励是根据他们工作量的大小来决定的。即使是右边那个矿工找到了有效区块，但左边那个获得了更多的奖励，因为他的工作量更大。找到有效区块的矿工并没有收到特别奖励。

矿池管理员如何分配奖励的方案有好几种，我们将会探讨一下最常见也是最简单的两种，也有其他一些方案被不同的矿池使用，但这两种基本上可以解释奖励方案之间的权衡选择。

工分分红

在这个模式里，管理员会对每一个超过特定区块难度的工分发放固定的奖励分红。在这个模式里，矿工在发送工分之后，管理员马上就会对其支付奖励，

而不需要等到整个矿池发现一个有效区块。

从某些方面来说，工分分红的模式对矿工是最有利的，他们可以确保每次发现一个工分的时候都有一定的收入，而管理员其实担当了所有的风险，因为无论矿工是否找到有效区块，他都必须按照工分支付奖励。当然，和其他模式相比，因为风险的增加，管理员也会收取更高的管理费用。

这个模式的问题是，矿工没有动力把有效区块提交给管理员。也就是说，即使把有效区块丢弃了，他们也会得到同样的奖励，但对整个矿池来说是个巨大的损失。一个恶意的管理员可以作为矿工参与另外一个矿池，用这个方法攻击另一个竞争对手，让他的矿池无法维持下去。

按实际比列分红

在这个模式里，不是按照工分分发固定分红，每个工分所能得到的分红，取决于整个矿池是否可以找到一个有效区块。每次找到一个有效区块，区块奖励（25 个比特币再加上交易费）会按照每个矿工的实际工作量按比例分配。

在这个模式里，矿工仍然会承担与矿池风险成一定比例的风险。但是如果矿池足够大，发现有效区块的概率波动会相当低。按实际比例分配的模式大大降低了矿池管理员的风险，因为只有矿池发现有效区块的时候才会支付矿工奖励。这也解决了工分分红模式的问题，矿工有动力把有效区块提交给管理员，因为只有那样他们的奖励才会被相应发放。

相比工分分红模式，这个模式略微增加了管理员的工作量，他要校验、计算和分配奖励。

矿池跳换

即使只有这两种矿池运营模式，我们可以看到矿工有动力去时不时地进行矿池跳换（pool hopping）。比如，一个按实际比例运行的矿池很快发现有效区块时，不管有效区块被发现的间隔是多久，管理员都会快速支付矿工奖励。

一个聪明的矿工可能尝试在挖矿周期的早期（也就是上一个区块刚刚被发现），在按实际比例分红的矿池中挖矿，这个时候的奖励可能相对比较高，然后只在周期

的后期切换（“跳”）到一个工分分红模式的矿池，这个时候按实际比例分红的矿池收益可能相对较低。这样导致的结果就是按比例分配的矿池可能无法运行。实际上更加复杂的方案可以防止这种矿工行为，比如“根据最近N个工分提交的结果才分配”是比较平常的做法，但即使这些方案也有可能诱发矿工跳换的行为。如何设计一个矿池方案以使其更好地防止这种行为，仍旧是一个有待解决的问题。

历史和标准化

矿池兴起于2010年比特币的GPU时代，并迅速变得十分受欢迎。道理很简单，因为它降低了矿工的概率波动风险。时至今日，矿池已经发展得十分先进。已经有很多矿池管理协议应运而生，甚至有人建议这些矿池管理协议应该被标准化，并且作为比特币本身的一部分。就像比特币的点对点网络协议一样，这些矿池协议也提供了一种特定的通信应用程序编程接口（Application Programming Interface，简称API），用于矿池管理员与每个矿工交流分派工作和矿工们递交工分给管理员。获取有效区块模版（getblocktemplate，简称GBT）就作为一种标准化的矿池协议放进了比特币改进方案（Bitcoin Improvement Proposal，简称BIP）之中。一种被称为层（stratum）的竞争协议，目前在实际中运用很广泛，就是一份BIP。不像比特币本身的协议，从事存在多个互不兼容的矿池协议没有造成太多的不便。每个矿池可以选择它们喜欢的协议，然后让市场来判定孰优孰劣。

有些挖矿机甚至把这些协定放进了硬件，但这最终会对限制这些矿机的灵活性有所限制。然而这使得购买矿机加入矿池变得异常简单。只需要把矿机插上电并连接上网络，选择一个矿池，然后这个矿机立刻就会接受该矿池的指令开始挖矿，并把电力消耗转变成收益。

51%的矿池

2015年早期的时候，绝大部分矿工都通过加入矿池来挖矿，只有很少的矿工还在单独挖矿。而在2014年6月，网络里最大的矿池GHash. IO，曾经变得如此巨大，其算力甚至超过了比特币全网算力的50%。主要是因为这个矿池给矿工优厚的奖励，以至于大家都想加入。

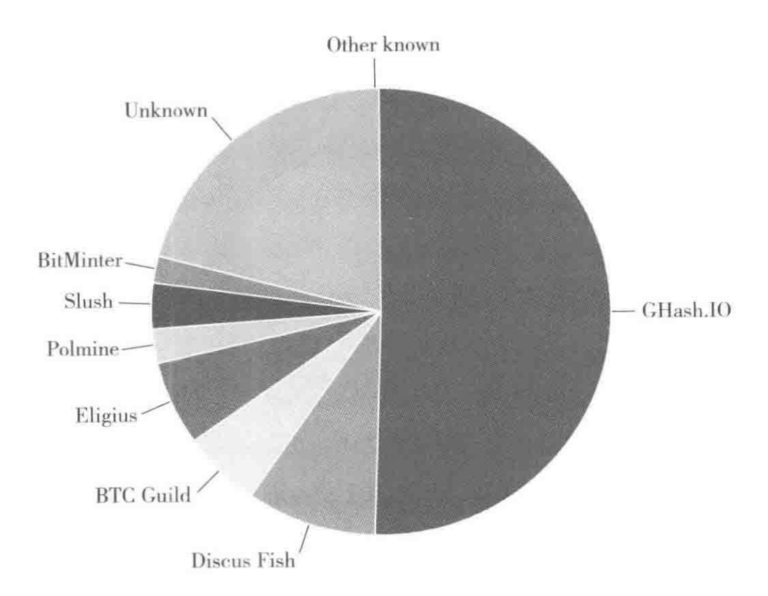


图 5.14（a） 矿池的算力分布

资料来源：blockchain. info（2014 年 6 月）

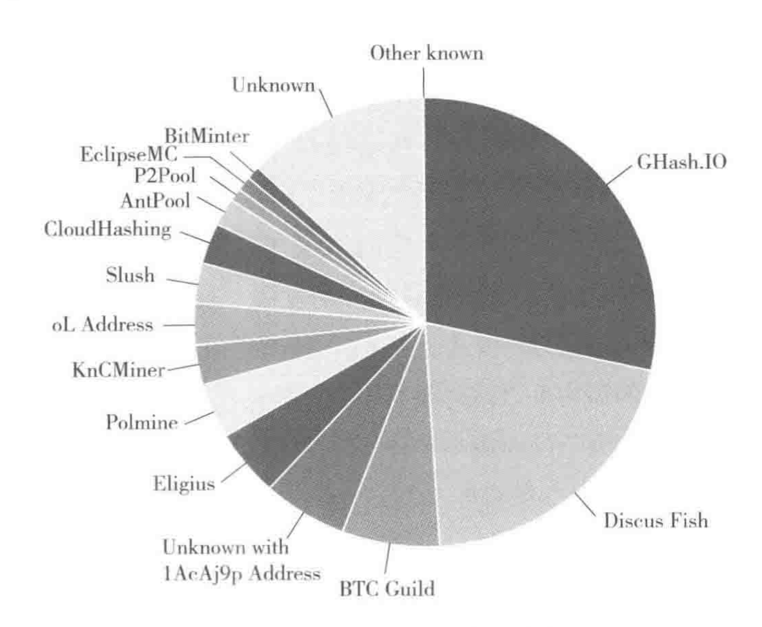


图 5.14（b） 矿池的算力分布

资料来源：blockchain. info（2014 年 8 月）

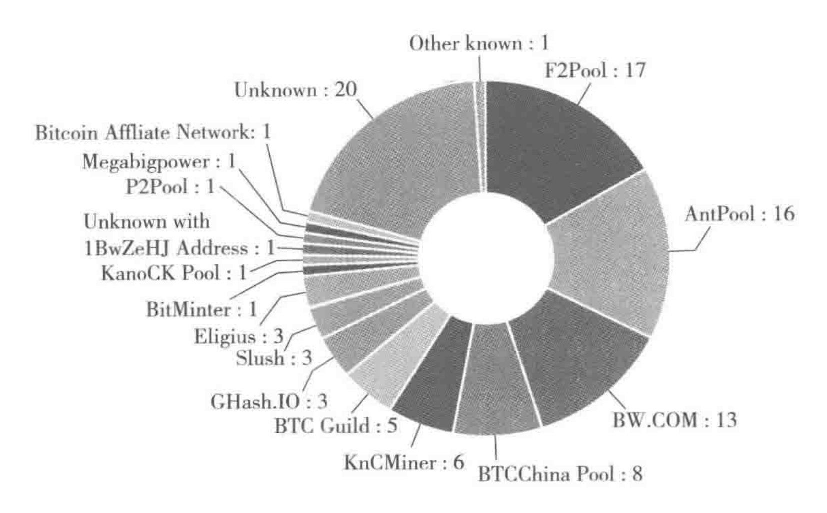


图 5.14（c） 矿池的算力分布

资料来源：blockchain. info（2015 年 4 月）

但这也是比特币社区一直所担心的，也导致了对 GHash 的反击。到了 8 月，GHash 不再接受新用户而主动下调了一些比例。即便如此，两个矿池依然掌控了整个网络一半左右的算力。

到了 2015 年 4 月，形势改变了许多，至少从表面上来看变得不是那么集中。但一个矿池掌控 51% 的算力依然是社区里一个令人担忧的问题。然而 GHash 遭受的负面的公众效应让很多矿池意识到这个问题，并尽量避免增长得过大。随着新的矿工加入市场，标准化的协议使得矿池之间的切换更加容易，矿池的市场份额一直在变动。矿池在长期如何发展，目前还不明朗。

无论如何，矿池有可能会掩盖这样一个事实：实际上的算力集中在几个大的挖矿机构手上，这些大的机构可以同时参与多个不同的矿池以掩盖它们的真实规模。这种做法被称为“洗算力”（laundering hashes，类似于洗钱）。因为矿池的原因，发现洗算力变得非常困难，这也使得外人无法知晓矿机的实际物理控制有多么集中。

矿池是有益的吗

矿池的好处在于矿工挖矿变得更加容易预测，也让小矿工更加容易参与。如果没有矿池的存在，挖矿效益上的概率波动会让小矿工承受不起。

矿池另外的一个好处在于，每一个矿池都有一个中心化的矿池管理员在网络中组装区块，所以网络更新变得更加容易。只要更新管理员的软件，即可更新所有矿池成员的软件。

当然矿池的一大问题是中心化管理。矿池管理员实际掌握了多大的算力是一个问题。当然，理论上一个矿工如果觉得管理员权力太大，可以自由地选择离开，但实际中有多少矿工会这样做还不清楚。

另一个坏处是减少了比特币网络上校验全部交易节点的数量（全节点）。以前，无论大小，所有矿工都必须自己运行一个全节点。他们要存储整个区块链，并校验每个交易。现在他们把这项工作交给了矿池管理员。这也是我们在第3章中提到的：整个网络中进行校验交易的全节点的数目在下降。

如果你对矿池的中心化模式感到不安的话，你可能会问：我们是否可以重新设计挖矿的流程，这样我们就不需要任何矿池，大家必须自己进行挖矿。我们会在第8章中探讨这个问题。

5.5 挖矿的激励和策略

我们在这一章花了很多篇幅讨论作为一个矿工的主要挑战：买到好的硬件、找到廉价的电费，然后尽快开始运行，然后期待一些好运气。不过在挑选一个区块开挖之前，每个矿工都需要做一些策略上的选择：

1. 需要包括哪些交易？矿工可以选择将哪些交易放进他的区块里。默认的规则是选择那些交易费比较高的交易。

2. 对哪一个区块进行挖矿运算？矿工可以选择在哪个区块上进行挖矿。默认的做法是在最长的那条区块链上继续挖下去。

3. 在同一高度的多个区块中做选择。如果两个不同的区块在同一时间被宣布发现，这就造成了一个区块的分叉，每个区块都是可以被延续下去的，因为它们都符合最长区块链原则。矿工必须选择其中一个区块接龙下去。默认的做法是选择最先被监听到的那一个区块。

4. 什么时候宣布新的区块？当矿工找到一个有效区块之后，他们要决定什么时候向比特币网络宣布这一个区块。默认做法是立刻宣布，但他们也可以选择等一下。

矿工其实面临很多决定。每个决定都有一个默认策略，直到这本书撰写之时，绝大多数的比特币客户端都是按照该默认策略运行的。非默认策略也有可能使得挖矿收益更高，很多人积极研究如何找到这样的策略。让我们来看看几种可能有别于默认策略的做法，这些做法可能使得挖矿收益更高。在接下来的内容中，我们假设一个运行非默认策略的矿工掌控一定的比特币网络挖掘市场份额，设为 α。

分叉攻击

最简单的攻击就是分叉攻击（forking attack），这是一个显而易见的获利方式——重复支付。一个恶意的矿工给一个受害者鲍勃发送了一些比特币来购买其服务和货品。鲍勃等到这笔支付交易被放进了最长链之后，甚至还等到了6个证实的时候确认支付安全之后，才开始发货或者提供服务。

现在这个矿工开始跳到前一个区块上开始重新挖矿——就是在那个包含他给鲍勃的支付交易区块之前的那一块。在这个分叉的区块链里，他插进了另一个替代交易——或者进行一个双重支付——把那些已经支付给鲍勃的比特币重新发送回自己的地址里（见图5.15）。

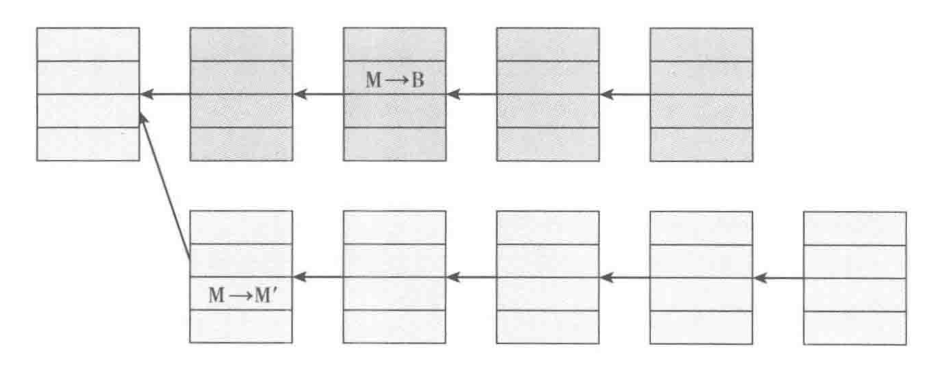


图5.15 分叉攻击

注：一个恶意的矿工给受害者鲍勃发送了一些比特币来购买其服务和货品。然后这个矿工进行了一个分叉攻击，创建了一个包含冲突交易的更长的分叉，在新的共识链中给鲍勃的支付就变成了无效的交易。

想要这个攻击成功，被分叉的区块链必须要覆盖当前最长的一条链，一旦这个情况发生，支付给鲍勃的交易就不再存在于共识的区块链里。如果这个矿工掌握占优势的哈希算力的话，也就是说 $\alpha > 0.5$，这种攻击就会成功。也就是说，即使有大量的随机变数，这个分叉最终会变成最长的一条链，也就是正当有效的共识链。甚者，因为这些币已经被用过了（在新的共识链上），这笔支付给鲍勃的交易永远不可能再回到区块链上了。

51%是必要的吗？如果 $\alpha > 0.5$ 的话，发动一个分叉攻击是很有可能发生的。但在实际中，用稍低的算力也可以发动一个那样的攻击，因为有类似于网络拥塞之类的其他因素。在主链上挖矿的其他矿工会因为一个正常的原因产生一些过时的区块——因为有网络延时。但是一个中心化的攻击者本身则不会有这个延迟，他可以进行更快速的通信并且生成更少的过时区块，这可能会节省1%甚至更多的算力。

拥有近乎50%算力的攻击者可能需要花很长时间才可能成功，因为有随机性。算力超过50%越多，攻击就会变得越容易也越有效。人们经常讨论51%的攻击，是因为51%是一个分水岭，超过51%的时候分叉攻击就会成为可能。实际上，这种攻击的成功概率是呈梯度变化的。

可操作的对策。我们不清楚分叉攻击在现实中是否一定成功。因为大家可以觉察到这个攻击，社区可以对此做出决定，即使分叉链更长也可以拒绝接受。

攻击和币值（exchange rate）。更加重要的是，这种攻击可以摧毁大家对比特币的信心，比特币的拥有者们就想要把资产转移出去，以至于比特币价格崩溃。因此，虽然一个具有51%算力的攻击者可能会在短期内利用双重支付进行欺骗并获得额外的收益，但是从长期来看，其实他们这么做造成的损失可能更大。

所以发动这种攻击的人其实是想通过打击信心来摧毁比特币。有点类似于邦德电影里的反派想要对诺克斯堡里的所有的黄金进行辐射污染，使其变得没有价值一样，这类攻击被称为金手指攻击（Goldfinger attack）。这种攻击者的目的可能就是摧毁整个货币，可能是由于他可以通过要么做比特币空头交易，要么拥有大量的竞争货币而获益。

通过贿赂来进行分叉攻击

通过购买足够多的矿机来控制大部分的算力，是一件非常困难而且昂贵的任务。但可能还是会存在其他简单的方法来进行分叉攻击：相比直接购买算力以获取超越所有其他人算力的昂贵做法，贿赂那些有能力的矿工来为你来工作也是可能的。

有几种贿赂其他矿工的方法。其中一个方法是"系统外的"（out of band）——可能找到一些矿工然后直接用现金来贿赂他们。当然一个更加聪明的办法是创建一个新的矿池，然后提供更好的奖励来吸引其他矿工来加入，即使矿池运行可能因此而亏损。虽然这种奖励不可能长期维持下去，但是可以维持足够长的一段时间直到可以发动一个成功的分叉攻击，然后获利。还有一种方法是在你的分叉区块链里留下足够多的"小费"，多到足以让其他矿工离开最长链来加入你的分链，矿工们希望你的链成为最长的链，这样一来他们可以收取你留下的小费。

不管是哪种贿赂的方式，核心思路都是一样的：有别于直接获得大量算力，攻击者去贿赂那些已经拥有算力的人，让他们帮助自己分叉出另外一条最长的区块链。

可能矿工们并不愿意去帮助一个攻击者，因为这么做会危害整个货币的价值，而他们已经在此之上投入了相当多的资金和矿机。从另一方面看，虽然矿工们作为一个整体可能希望保持货币的价值，但是他们可能做不到一致行动。个别矿工可能会因为短期利益，将个人利益置于集体利益之上。从经济学的角度来看，那就是个经典的"公地悲剧"了。

这些假想在现实中未曾发生。贿赂攻击是否可行，这依旧是一个悬而未决的问题。

临时保留区块攻击

假设找到一个区块之后，默认的做法是你会立刻向全网宣布找到的区块。但是如果你想进行一个临时保留区块攻击（temporary block-withholding attacks），

你也可以不立刻宣布，然后在这块上面继续挖矿，期望你可以在其他矿工找到下一个区块之前连续找到两个有效区块，在整个过程中秘密地保留你所发现的区块。

如果你已经拥有两个公共区块链上超前的秘密区块，那么全网剩下的矿工所做的挖矿努力都会被浪费，其他的矿工都会在他们认为最长的链上继续挖矿，一旦他们宣布他们找到了一个有效区块，你可以立刻宣布你所秘密保留的两个区块，这样你的区块链立刻变成了最长的有效链，而其他人辛苦挖出来的区块马上就变成了一个孤块而被丢弃（见图 5.16），你的这种行为被称为自私挖矿（selfish mining）。通过使网络上的其他矿工浪费算力计算出来的区块瞬间过期，可以有效地增加你的挖矿获利。

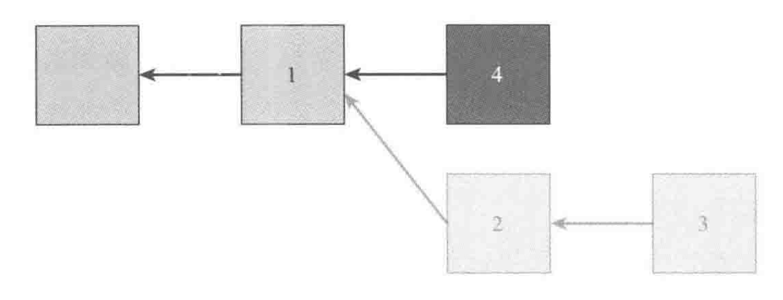


图 5.16　自私挖矿图示

注：图中显示了其中一种攻击方式。(1) 攻击之前的区块链。(2) 攻击者挖到的区块，保留着，在此之上继续挖矿。(3) 攻击者运气很好，在全网其他矿工之前发现了第二个区块，并继续保留。(4) 非攻击者找到了一个区块，并进行广播。攻击者立刻广播他所保留的两个区块，使得区块 4 变成了孤岛，浪费了其他人之前所用的算力。

这里面的关键是你需要运气好到连续发现两个区块，风险在于你只领先了一个区块，其他人就已经向网络宣布发现了一个有效区块。如果这种情况发生，你必须立刻宣布你的秘密区块，这叫造成了一个区块的分叉，每个矿工都需要选择哪一个区块继续挖下去。当然，你希望大部分其他矿工最早监听到你的区块并在上面继续挖矿。由于这种攻击的有效性严重依赖你赢得这个竞赛的能力，所以网络位置至关重要。你可以尝试跟所有的节点建立链接，以使得你的区块可以第一个到达其他的节点。

假设只有50%的机会可以赢得这个竞赛，在 $\alpha > 0.25$ 的情况下，自私挖矿可以比默认策略更有收益。如果 $\alpha > 0.333$，即使你输掉每一个这种竞赛，仍然可以获得更高的收益。这种攻击的存在是令人震惊的，原来大家都相信如果没有很大的算力——比如 $\alpha \leqslant 0.5$——不会有比默认策略更有利的挖矿策略。所以，即使某个矿工控制的算力低于50%，也是有可能通过切换到其他的挖矿策略来获取更多的收益。

到2015年为止，临时保留区块攻击仅仅是理论上的，在实际中并没有观察到这类攻击事件，自私挖矿则很容易被检测到，因为这种策略会增加同时宣布区块的概率。

黑名单与惩罚分叉攻击

如果一个矿工想把一个来自地址X的交易列入黑名单，换句话说，他想冻结从该地址出来的钱，让这些钱变得不可用。或许他想用这个办法来敲诈勒索一笔钱，或许他们之间有仇，还有可能是政府执法部门认为那些地址有问题，需要矿工的配合来冻结这些币。

传统观点都认为在比特币里这种黑名单没有办法有效施行。因为即使有些矿工会拒绝把交易放进区块链里，其他一些矿工可能会。如果你真的想把一笔交易列入黑名单，你可以尝试其他一些更加激烈的手段，比如，惩罚分叉（punitive forking），你可以宣布拒绝在包含来自该地址的交易的区块链上工作。如果你拥有大部分市场运算能力，那应该足以保证这个黑名单上的交易永远不会被公布。确实，在这种情况下，其他矿工很有可能不会再试图把这笔交易放入区块链里，因为这么做有可能使得他们自己的区块链被分叉，这会导致他们发现的区块被删除。

羽量级分叉

如果没有很大的算力，上述的几个分叉攻击在现实中都不太可能实现。如果你宣布拒绝接受包含某些特定交易的区块链，但这条链被网络上的其他矿工所接受并形成最长链的话，你就会发现自己被永远排除在共识链之外（这就是

一个硬分叉），所有你做的挖矿工作统统浪费了。更加糟糕的是，黑名单上的交易仍然存在于最长的区块链上。

换句话说，考虑到还有其他矿工的存在，用惩罚分叉把特定交易放入黑名单的手段并不可靠。然而，有另一个更明智的方法可以做到这一点。与其一看到从地址 X 里出来的交易就宣布你会进行永久分叉，不如宣布你将会尝试分叉，但过一段时间你可能会放弃封杀的尝试。例如，你可以宣布：当 k 个区块证实了从这个地址出来的交易是正当的时候，你便会回到最长链。①

如果你在一个区块证实后便放弃，把那笔从地址 X 出来的交易成功封杀的概率是 α^2。原因是你必须要在其他矿工找到下一个区块之前找到连续两个区块，这样才能成功地丢弃那个包括地址 X 交易的区块。α^2 是你连续找到两个区块的概率。

α^2 这个概率看上去不是很好。就算你掌控了 20% 的全网算力，也只有 4% 的成功概率来封杀那笔你不希望出现在区块链上的交易。但这已经不错了，至少你还有可能说动其他矿工来加入你。只要你把你的计划公开了，其他矿工便会知道：如果他们胆敢把这个来自地址 X 的交易加入自己的区块，便有 α^2 的可能会丧失自己已经发现的区块［被你的羽量级分叉攻击（feather forking）所消灭］。只要他们不是有很强的主观意愿把这个交易包括进来并且这个交易没有很高的交易费，他们可能更愿意规避那 α^2 失掉过往挖矿奖励的风险，而不是获取那笔交易费。

这就演化为：其他挖矿者经过理性的思考，将决定加入你对 X 地址的封杀行动，这样你便可以成功地封杀 X 即使 $\alpha < 0.5$。所以这个攻击要想成功，重点在于确保其他矿工相信你将会进行分叉攻击。

逐渐转移到用交易费来奖励挖矿

直到 2015 年，交易费还不是那么重要，因为区块奖励在矿工总收入里占比超过 99%。但每 4 年，区块奖励就会被减半，最终区块奖励将会变得很低，低

① 给自己留条后路，见机行事。——译者注

到交易费变成了矿工的主要收入来源。届时矿工会如何应对还属未知。他们会不会更加激进地要求实行最低交易费？矿工会不会联合起来逼迫比特币网络实行最低交易费制度？

未解的问题

总结来说，理论上矿工可以自由地选择挖矿的策略，但在实际中我们观察到的是大部分矿工都选择了默认策略来挖矿，虽然没有完整的模型可以证明默认策略（default strategy）就是最佳的。在本章中，我们讨论过几个特定案例，有大量算力的矿工有可能执行非默认策略来获取更大的收益。在挖矿策略上，实践是领先于理论的。在实践中，大多数矿工还是选择了默认策略，而且比特币运行得也很好。但是，从理论上，我们还无法论证这是一个稳定的机制。

默认策略能否在实际运行中一直保持有效，对于这一点我们也没有把握。比特币运行所依赖的现实条件也一直在改变。矿工们变得越来越中心化和专业化，整个系统的算力也越来越大。另外，从长期来看，比特币的奖励将从固定的挖矿奖励为主转变为交易费为主。我们真的不知道这将会如何演变，基于博弈理论对此进行预测也是一个非常有趣的前沿研究领域。

延伸阅读

关于挖矿硬件演变的一篇优秀论文：

Taylor, Michael Bedford. “Bitcoin and the Age of Bespoke Silicon.” In *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*. Washington, DC: IEEE Press, 2013.

关于比特币和加密货币的知识系统化文章，特别是第三部分关于稳定性：

Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. “Research Perspectives and Challenges for Bitcoin and Cryptocurrencies.” Presented at the 2015 IEEE Symposium on Security and

Privacy, San Jase, CA, May 2015.

一份完整的分析2011年不同矿池激励机制的文章（有些信息有点过时，但是整体上还是值得参考）：

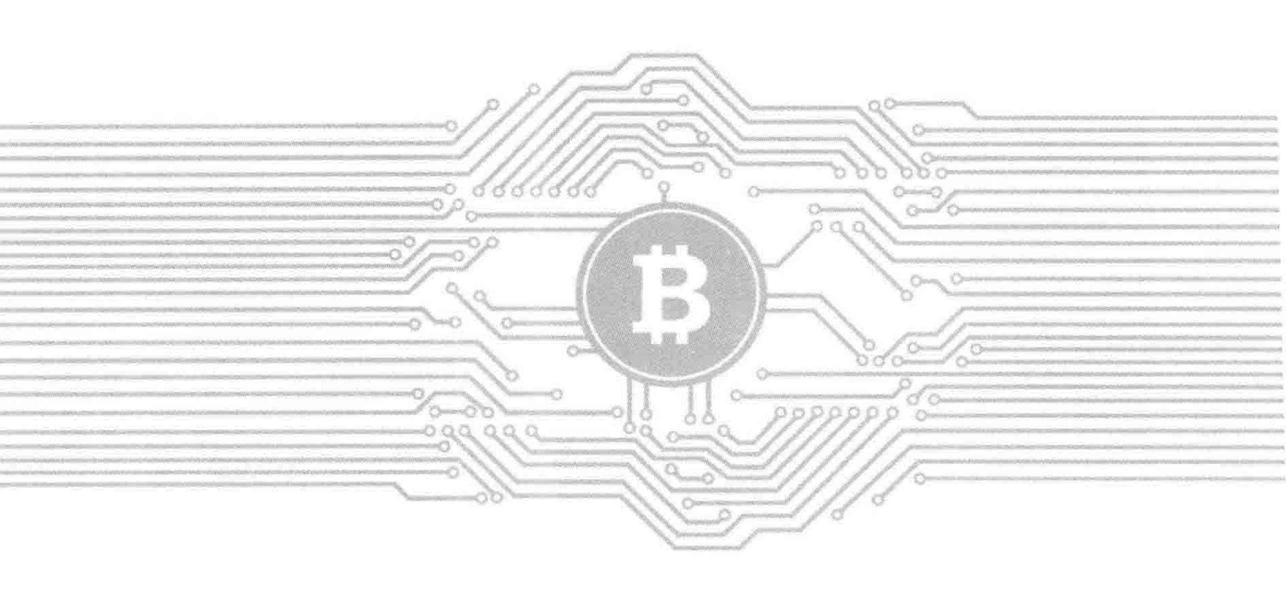
Rosenfeld, Meni. "Analysis of Bitcoin Pooled Mining Reward Systems." arXiv preprint arXiv: 1112.4980 (2011).

几篇研究挖矿策略的论文：

Eyal, Ittay, and Emin Gün Sirer. "Majority Is Not Enough: Bitcoin Mining Is Vulnerable," In *Financial Cryptography and Data Security.* Berlin and Heidelberg: Springer, 2014.

Kroll, Joshua A., Ian C. Davey, and Edward W. Felten. "The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries." In *Proceedings of the Workshop on the Economics of Information Security 2013*. Berlin: Springer-Verlag, 2013.

Eyal, Ittay. "The Miner's Dilemma." Presented at the 2015 IEEE Symposium on Security and Privacy, San Jose, CH, May 2015.



第6章

比特币和匿名性

BITCOIN
AND
CRYPTOCURRENCY
TECHNOLOGIES
A Comprehensive Introduction

比特币是一种安全并且匿名的数字货币。

——维基解密捐款页面

比特币并不会帮你躲避国家安全局的窥探。

——英国有线

比特币最有争议性的一个特性，就是匿名性。首先，比特币是匿名的吗？从上述两种完全相反的论调可以看到，人们对比特币匿名性存在理解上的困惑。其次，我们需要**加密数字货币**（crypto-currency）做到完全匿名吗？匿名性有好的地方，也有不好的地方，进而引申出一些基本的问题：加密数字货币的匿名性对持有者有益处吗？对社会有没有好处呢？有没有一种方法，可以让匿名性只发挥积极正面的作用，而不用担心它的负面作用呢？

这些问题很难回答，因为它们取决于人的道德价值观。本章中，我们并不会回答这些问题，即使我们仍然会探讨匿名性的优劣。我们将主要研究探讨各种技术特性，其中有一些已经是比特币所具备的，还有一些是为增强比特币的匿名性，而建议比特币应该添加的技术特性。我们也会对其他一些具备不同匿名技术特性的加密数字货币进行研究。这些技术也会带来新的问题，比如它们能在比特币或者其他加密数字货币中正常工作吗？接受这些特性有多困难？接受这些特性需要对现有的技术功能做出哪些取舍？

6.1 匿名的基础知识

匿名的定义

在讨论比特币是否具备匿名性之前，我们需要对匿名做一个定义。我们需要准确地理解到底什么是匿名，以及它与其他一些相似术语的关系，比如隐私。

匿名和化名（pseudonymity）

在其他你可能更熟悉的情形下，说明匿名和单纯的化名的区别可能更容易。其中一个很好的例子就是在线论坛，在一个类似于红迪（Reddit，美社交新闻网站）的论坛中，你会选择一个自己常用的化名在一段时间内和系统进行交互。你也可以创建多个化名，甚至每一条评论都用一个新的名字，但那样会很麻烦，绝大多数人都不会这样做。所以在红迪这样的论坛里，用户通常都会使用化名，但又不是完全匿名。相反，在4Chan（综合型讨论区）这种在线论坛里，用户通常用匿名来发帖，并且不带有任何标识性属性。

从字面上理解，匿名的意思是“没有名字”。当我们尝试用这个定义来说明比特币的匿名特征时，会有两种不同的诠释：在交易的时候不使用真实的姓名，或者在交易的时候完全不使用任何名字。比特币是否具备匿名性，这两种释义会带来两种完全不同的结论。比特币的地址是**公钥哈希值**（hashes of publickeys），在与比特币系统交互过程中，使用者不需要使用真实的姓名，但是需要使用公钥哈希值来作为交易标识。因此，按照第一种对匿名的释义，比特币是具有匿名性的，因为使用者不需要使用真实的姓名。然而，如果根据第二种释义，比特币并不具备匿名性，因为交易中必须使用的地址是一种虚假标识。在计算机科学语言中，这种不用真实姓名而使用一种特定标识的折中做法被称为化名。

你可以根据你的需要，随意创建出任意多个比特币地址，考虑到这一点，你可能会思考比特币的地址是否真的是虚假标识，因为你可以创建无穷多的化名，正如我们将会看到的，这并不能让比特币具备匿名性。

在计算机科学中，匿名指的是具有无关联性（unlinkability）的化名，无关联性是一种针对特定攻击者的能力而定义的属性，从直观的意思看，无关联性意味着如果一个用户和系统重复进行交互，从特定攻击者的角度考虑，不同的交互行为之间应该无法互相关联。

比特币是具备化名性的，但如果你的目的是要求绝对隐私，那么这种匿名性还不够。区块链技术是一种公开的账本系统，任何人都可以查询包含了给定地址的所有比特币交易。如果有人可以用你的比特币地址链接到你的真实身份，那么所有你的比特币交易记录——不管是过去的、现在的，还是未来的——都能关联到你的真实身份。

更加糟糕的是，把比特币地址和真实身份链接起来并不困难。如果你跟某一种比特币业务有关联 ——不管是一个在线电子钱包服务，还是其他接受比特币的商家或交易所，通常你都需要提供你的真实身份以完成相关交易。比如，交易中通常都需要你提供信用卡信息，或者商家需要你提供真实地址以便送货上门。

或者你去咖啡馆喝杯咖啡，然后用比特币来支付。由于你已经出现在店铺里，咖啡师对你的身份特征已经有了非常多的了解，即使他们没问你的真实姓名。你的物理标识就这样和你的某个比特币交易联系到了一起，从而可以由这个地址追踪到你所有的其他比特币交易。这显然不具备匿名性。

旁路攻击（side channels）

即使没有发生直接的关联，因为侧面渠道或者一些间接的信息泄露，你的匿名身份也可能会被**暴露**（deanonymized）。举个例子，某个人看过一个匿名的比特币交易记录，并且注意到了这个比特币交易的活跃时间，那么他就可以将这个时间信息与其他公开可获得的信息关联到一起。可能他会注意到在同一个时间某一个推特用户也出于活跃状态，那么就可以建立一个匿名的比特币资料和一个真实世界的用户（至少是一个推特账号）之间的关联。很显然，这样的匿名性并不能

保障隐私或者绝对匿名。要想做到绝对匿名，我们需要更强的无关联性属性。

无关联性

为了更扎实地理解比特币范畴中的无关联性特征，我们可以列举一些在比特币交易中无关联性所需要的关键属性：

1. 同一个用户的不同地址应该不易关联。
2. 同一个用户的不同交易应该不易关联。
3. 一个交易的交易双方应该不易关联。

上述第一条和第二条很容易理解，但第三条比较微妙。如果你把一笔“支付”诠释成一个比特币交易，那么第三条属性很明显就是一个伪命题。每一笔交易都有输入和输出，这些输入和输出都不可避免地会记录在区块链网络中，并且公开地关联在一起。然而，我们所指的支付并不仅仅是一笔比特币交易，而是指任何一种从发送者到接收者的传输比特币的行为，这种行为可能会涉及一系列的间接迂回交易。我们需要确保，通过查询区块链上的信息将发送者和最终的接收者关联在一起，是不可行的。

匿名集

即使我们对支付做出更加宽泛的定义，第三条属性看起来也比较难以实现。比如说，你支付一定数量的比特币来购买某个商品，并且通过迂回曲折的形式发送了这些比特币，其他人通过查看区块链上的信息，还是可以推断出某个比特币地址上减少了一定数量的比特币，而另外一个地址上增加了差不多相同数量的比特币（可能会扣除相应的交易费用）。此外，尽管传输是通过迂回曲折的路径，初始发送方发送比特币和最终接收方接受比特币基本上发生在同一个时间段，因为商家不太愿意接受延迟付款。

基于这样的困难，我们通常并不试图在系统中，对所有可能的交易或者地址都实现完全的无关联性，而是去实现更有限度的无关联性。想象一个特定的攻击者的情况，你的交易匿名集（anonymity set）是指该攻击者无法把你的交易从其中分辨出来的交易集合。即使该攻击者知道你完成了一个交易，但是他也仅仅知道这

个交易是某一个合集中的一个，但并不能确定是哪一个。我们只需要努力去最大化这个匿名合集就可以了——在这个合集中，我们可以隐藏我们的地址或交易。

统计匿名集是很微妙的，由于这样的匿名集是针对某一个或者一组攻击者而定义的，所以你首先需要具体定义你的对手模型是怎样的。你必须要仔细思考对手已知和未知的内容，以及我们需要隐藏的内容——也就是说，如果要达到匿名性目标，交易中的那些信息是不能被对手知道的。没有成熟的公式告诉你该怎么做，需要根据每一个具体情况仔细分析相对应的协议和系统。

污点分析（taint analysis）

在比特币社区中，人们通常根据直觉推断匿名性，而非严格的定义。污点分析就是一种非常流行的方式：这是一种推算两个地址相关性的方法。如果地址 S 发送出的比特币总是地址 R 接收，那么不管是直接抵达，还是经过了多少中间地址，S 和 R 则被定义为具有高分污点。污点分析的计算公式，适用于多个输入和（或）输出的交易，并且确定如何分配污点的规则。

遗憾的是，污点分析也不是一个衡量比特币匿名性的好方法。它只是简单地认定对手在使用相同的计算方式在关联成对的地址。稍微聪明一点的对手会使用不同的技巧，比如查询交易时间，或者本章后面我们会讲的利用钱包软件的特性。所以，污点分析可能只会显示你在某种特定情况下具备的匿名性，而事实上可能并不准确。

为什么需要匿名性

了解了匿名性意味着什么之后，在我们进入更深入的探讨之前，让我们来回答一下有关匿名性的根本问题：为什么人们需要匿名性？拥有匿名性的货币有哪些伦理道德方面的意义？

在区块链货币中，所有交易都被记录在一个公共账本上，也就是说，这些记录都是公开的，并且可根据相关地址进行永久追踪，因此，你的比特币交易的隐私保护可能会比传统的银行更糟糕。如果你的真实身份被关联到一个比特币地址，那么你就完全失去了所有交易的隐私——不管是过去的、现在的，还

是未来的——只要是和这个地址相关联的交易。由于区块链是公开可用的，确切地说，任何人都有可能在你不知道的情况下识别你的身份。

考虑到这一点，我们可以认定两种不同的需要匿名加密数字货币的动机：第一种是，达到我们习惯的传统银行给我们的隐私保护级别，降低公共区块链所带来的信息暴露风险；第二种是，要超越传统银行给我们的隐私保护级别，进一步开发数字货币，使其从技术上实现任何人不能轻易追踪参与者。

匿名性的道德问题

我们有很多非常重要的理由（虽然经常被忽视）需要匿名性，传统货币已经让我们对这种匿名性习以为常。大多数人都不愿意与朋友以及同事分享他们的工资收入状况，如果你的工资是用区块链比特币支付的，而且该区块链上的地址很容易被识别，那么我们只需要关注每个月定期的大额支付记录，就能很容易地推断出你的工资情况。企业组织也有非常重要的财务隐私顾虑，例如，一家电玩游戏生产商被发现在区块链上支付给了一个分包商，而这个分包商是专门生产虚拟现实眼镜的，那也意味着它们要推出的新游戏可能会被公众（也包括其竞争对手）提前知晓。

然而，真正合理的顾虑是，匿名加密数字货币可以被用作洗钱或者从事其他非法活动。即使加密数字货币的交易本身可能是匿名的，数字货币和法定货币之间的接口却无法做到匿名，这可以说是一个好消息。事实上，我们将在下一个章节讨论中看到，这些数字货币和法定货币之间的兑换和流动被严格监管着。因此加密数字货币也不是洗钱或者其他金融犯罪的灵丹妙药。

然后有人可能会问：我们是否可以设计出一种技术，这种技术只允许使用好的匿名，而禁止坏的匿名呢？事实上，计算机安全和隐私方面的研究者一直在探索这个问题的答案，遗憾的是，从来就没有可行的方案，因为我们用道德标准分辨出的好坏，对计算机技术来说，是没有办法区分的。在比特币系统中，如何让矿工按照道德标准去判定哪些交易应该被发布到区块链中，目前还是无法实现的。

我们的观点是，启用匿名加密数字货币是有潜在好处的，这也是其得以存在的原因，同时，对于系统的技术匿名属性和在使用货币时应该遵守的法律规

范，我们应该加以区分。匿名加密数字货币并不是一个完美的解决方案，但可能是最优的一种折中方案。

匿名化和去中心化

在本章中，我们会看到很多次有关匿名化和去中心化的讨论，通常这两点是相互矛盾甚至冲突的。回忆一下前言部分提到过的乔姆（Chaum）的电子现金系统，它在一定意义上实现了完美的匿名性，不过必须依赖一个中央权威机构——银行的盲签名协议。设想一下，这样的协议的去中心化将会非常困难。如果我们强制进行去中心化，就必须要有一种能够追踪交易并且防止双重支出的机制。交易的这种公开追踪特性，就是对匿名化的一种威胁。

Tor①

如何处理一个有好坏两面性的技术，比特币并不是唯一一个面临这种道德上两难选择问题的技术。另外一个匿名性设定有争议的系统是 Tor，一个匿名通信网络。

一方面，Tor 的用户只是一些普通人，想保护他们在网络上不被跟踪。其用户包括记者、社会活动家和持不同政见者等，他们追求在互联网上的言论自由而不用担心政治迫害。同时，执法部门的探员也使用这个系统，监控在线嫌疑犯，而不会泄露他们的 IP 地址（毕竟，大家都知道 IP 地址是根据不同的组织机构来分配的，包括执法部门）。很显然，Tor 有很多应用场景是我们从道德上认可的。另一方面，它也还有一些不好的应用场景：有人用其运作僵尸网络来控制一些被感染的电脑，进而用来传播淫秽等非法图片。

要想从技术角度去区分这些使用场景，基本上是不可能的。Tor 的开发者以及社区一直在尝试解决这个难题，全社会也在努力尝试。我们似乎可以总

① The Onion Router，洋葱路由，用户通过 Tor 可以在互联网上进行匿名交流。——译者注

结一下，整体而言，这样的技术利大于弊，实际上，Tor 主要的投资方之一就是美国政府，其投资兴趣在于，Tor 可以让不同政见者在互联网上发表自由言论，而不用担心政治迫害。同时，执法部门也勉强接受 Tor 的存在，甚至还开发了使用这个平台的方法。FBI 也经常会查封一些在“黑暗网络”上非法传播色情图片的网站，即使这些网站也使用了 Tor 的平台，多数情况是，这些网站的运营者被绳之以法。我们必须要记住，技术仅仅是工具，而那些犯罪分子生活在现实世界里，并可能留下各种犯罪证据，或者在使用这些工具的时候犯一些人为错误。

在本章 6.5 节中，我们会探讨零币（Zerocoin）和零钞（Zerocash），这是一种匿名化并且去中心化的加密数字货币系统，有点类似于乔姆的电子货币（ecash），但是因为上述两个限制，它们需要找到合适的方法，以解决棘手的加密问题。

6.2 如何对比特币去匿名化

我们已经强调过很多次比特币仅仅是一个化名系统，所以你所有的交易记录或者交易地址很有可能被关联在一起。让我们再进一步讨论这种关联是怎么发生的。

图 6.1 展示了维基解密的捐款页面上的一个片段（包括本章开头的时候就引用的一段），请注意那个在比特币地址旁边的刷新按钮。可能你会期望通过点击这个刷新按钮，就会把接收捐款的地址换成一个全新生成的地址。类似地，如果你刷新页面或者关闭这个页面，重新再打开的时候，这个地址也会重新生成，而且是之前没有出现过的。这是因为，维基解密需要保证每接收一笔新的捐款，都会对应一个新创设的仅用于此笔捐款的公钥。维基解密这么做就是最大限度地利用了可以创建新化名功能的作用。这事实上就是比特币钱包实现匿名性的最好途径。

比特币是一个安全的匿名数字货币系统，不容易被追踪，并且是一种有别于其他捐款方式的、更加安全和快速的方法。你可以发送比特币到以下地址进行捐款：

13DFamCvSxG8EG16VyXzdpfqxyooifswYx 

图6.1　维基解密的捐款页面的一个片段

注：请注意那个在比特币地址旁边的刷新按钮。维基解密遵循了为每一笔捐款生成一个新接收地址的比特币匿名化的最佳实践。

你可能觉得这些不同的地址一定是无法关联的，维基解密收到的不同的捐款是完全分开的，并且推测它们可以分开使用每一笔捐款，但事实并非如此。

关联性（linking）

假设爱丽丝想要去买一个茶壶，价格是8个比特币（可能根据2015年的比特币价格，实际情况应该是8分比特币，1个=100分）。进一步假设，她的比特币分在三个不同的地址里面，分别有3、5和6个比特币。实际上，爱丽丝没有一个比特币地址有足够的8个比特币，她必须要把两个输出合并成一个单体输入，以支付给店铺。

隐形地址

假设鲍勃要通过他的网站还有广告牌来宣传他的捐赠地址。现在还没有任何方式可以将一个不同的地址显示给每个用户，必然地，接收现场捐款的这个地址会很容易连接到鲍勃的站点。

能够利索地解决这个问题的办法，是利用隐形地址（stealth addresses）。它允许收件人鲍勃发布一个静态“永久”的地址，任何发件人（比如爱丽丝）由此可以派生出新的地址，该地址的私钥只有鲍勃知道。

这是如何做到的呢？回忆一下椭圆曲线数字签名算法（ECDSA）中的公钥的函数形式是g_x，其中的x是私钥，地址函数是H（g_x）。为了启用隐形地址，鲍勃需要广告公钥本身，而不是长度更短的哈希值。然后，爱丽丝可以选

取一个随机数值 r，计算（g_x） r = g_{xr}，并且将钱汇给这个公钥。如果爱丽丝能够单独将数值 r 发送给鲍勃，鲍勃就可以计算出正确的私钥 xr，将汇到公钥 g_{xr}的钱花掉。

这种方法并不完美，因为爱丽丝需要将数值 r 发送给鲍勃，而且还假定即使鲍勃不在线，比特币交易照样运行。为了解决这个问题，还有更复杂的协议，让爱丽丝能够有效地将数值 r 嵌入比特币交易本身。随后，鲍勃可以扫描区块链，检测针对他的交易，并恢复私钥。黑暗钱包中使用了这种方法，该钱包设计时旨在增强隐私，并且类似的想法在加密签名（CryptoNote）这种另类币中有所使用。

那么问题来了，这笔交易会在区块链网络里有一个永久的记录，任何看到这个记录的人都可以推断，这两个输入型交易很有可能是由同一个用户控制的。换句话说，共享型消费，成为不同输入地址联合控制的证据。当然也可能存在例外，有可能爱丽丝和鲍勃是同寝室的朋友，决定一起联合购买这个茶壶，并且分开付款。但是，大体来说，共同输入基本上意味着共同控制。

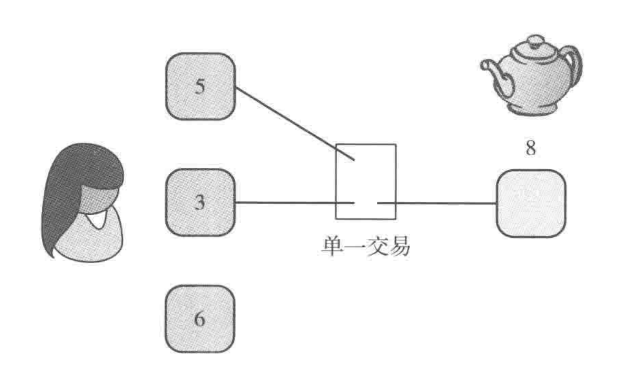


图 6.2　多地址输入交易

注：为了支付购买茶壶的钱，爱丽丝从两个不同的比特币地址创建了一个单一交易。这样爱丽丝就暴露了一个事实，即一个个体控制了两个不同的地址。

还不仅仅如此，攻击者可以重复上述过程，从而一步一步将这个个体所进行的所有交易关联起来。如果另外一个地址也关联到了爱丽丝用于交易的两个

地址之一，那我们就知道所有三个地址都属于同一个个体，我们可以因此建立一个**地址簇**（clustering of addresses）。一般来说，如果一个新地址的输出，和该地址簇中的任何一个已知地址被一起花费，那么这个新的地址也将会被加到该地址簇中去。

在本章6.4节，我们将会探讨一种叫合币（CoinJoin）的匿名技术，该技术的工作原理打破了上述设想。但是到目前为止，对没有使用特别匿名技术的普通比特币钱包用户，这种簇技术还是非常有效的。接下来，我们很快将会讨论如何把这种地址簇关联到现实世界的身份。

零钱地址（change address）的随机化

早期版本的比特币类库［Bitcoin-Qt library，现在又称为比特币中心（Bitcoin Core）］存在一个缺陷，对有两个输出地址的交易，它总把存放零钱的输出地址放在第一个，这意味着很容易分辨出很多交易中的零钱地址。这个缺陷在2012年得到了修复，但是重点在于：钱包软件在保护匿名性中扮演着非常重要的角色，如果你正在开发钱包软件，你需要格外注意很多陷阱，尤其是你需要保证零钱地址的位置应该永远是随机的，以避免为攻击者留下太多信息。

再回头看一下我们的例子。假设这个茶壶的价格从8个比特币上涨到了8.5个比特币，爱丽丝发现，未用完的支出账号里无法再组合生成恰好可以支付这个茶壶所需要的金额了。取而代之的方案是，爱丽丝利用交易可以有多重支出的特性，如图6.3所示，支出的其中之一就是茶壶店铺的接收地址，而另外一个则是爱丽丝自己的“找零”地址。

现在从其他人的视角来看这笔交易，他们可以推断出这两个输入地址都属于同一个用户，他们甚至可能怀疑其中一个支出地址也属于这个用户，但是无法知道具体是哪一个。事实上0.5个比特币虽然比其他的支出小，但是并不意味着这是一个零钱地址，爱丽丝可能有10 000个比特币参与了交易，其中她支

付了8.5个比特币用于购买茶壶，而把剩余的9 991.5个比特币找零退回给了自己。在这样的场景中，更大的输出才是实际上的找零地址。

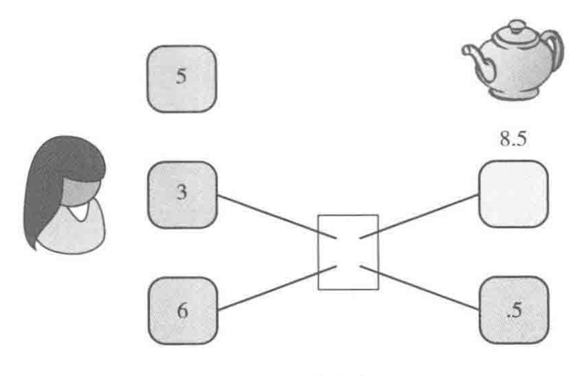


图6.3　零钱地址

注：为了支付购买茶壶的钱，爱丽丝创建了一个交易，这个交易中的一部分比特币去了商家的钱包，而剩余的部分作为零钱退回给了她自己。

另外一种更好的解释是，如果茶壶只需要花费0.5个比特币，由于不管是3个比特币还是6个比特币的输入地址，都足够用来支付了，爱丽丝根本不需要创建两个不同输入组成的交易。但是，选择哪种交易方式完全取决于通常使用的钱包软件的特性，即使不是非常有必要，钱包（或者是用户）还是可以随意组合不同交易地址中的比特币来完成支付的。

惯用法则

这种类型的实施细节被称为“惯用法则”（idioms of use）。2013年，一组研究人员发现了大多数钱包软件都在使用的惯用法则，并推导出一种用来鉴定零钱地址的强大方法。具体而言，钱包在有需要的时候都会生成一个全新的地址，因为这种惯用法则的使用，这些新的地址通常都是从来没有在区块链网络出现过的。换句话说，非零钱地址通常都不是新地址，而是已经在区块链网络里出现过的，那么其他人就可以利用这个特性去分辨零钱地址，并把它和输入地址相关联。

依赖惯用法则来推测零钱地址可能会出错。事实上，零钱地址是新地址这一特性不过恰巧是钱包软件的一个特性。在2013年研究者测试的时候发现确实

是这样，现在可能还是，但也有可能不再如此了。用户可以选择覆盖掉原来的默认设定，最重要的是，当对手了解了这种技术的时候就可以很容易回避，即使是在2013年，研究者也发现这种特征也会经常产生误报，按照这种规则可以归到一个簇的地址，不一定是属于同一个个体的。研究者声称，他们需要大量的人工监督和干预才能去除这些误报。

关联真实世界的身份到地址簇

在图6.4中，我们可以看到，米克尔·约翰（Meikle John）等人是如何利用惯用法则这样的启发式算法来聚类比特币地址的，但是这种簇没有标签——也就是说，我们还没有关联一个真实的身份到这个簇。

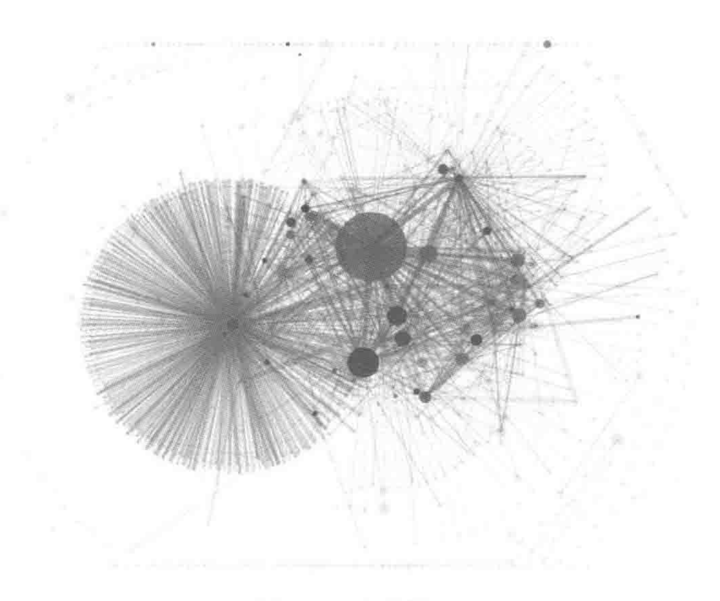


图6.4　地址簇

注：摘自2013年的一篇论文“一把比特币：寻找支付特征”。在一组没有姓名的用户中，研究者将联合支付的地址和全新的零钱地址归类到一个比特币地址簇。图中，圆形的大小表示流入这些地址簇里的货币数量，每一条线则代表一个交易。

我们可以根据所了解的比特币经济情况来做一个合理的推测。回到2013年，门头沟公司曾经是最大的比特币交易所，所以我们可以猜测图中较大的圆圈代表的就是该交易所控制的地址，我们可能也注意到，图中左侧的深色的圆

圈代表了很小的比特币总量，但同时又有非常大的交易量，这个特性很符合叫作中本聪之骰（Satoshi Dice）的在线比特币博彩游戏，这个游戏中，你可以发送微小量的比特币作为赌注。总的来说，这并不是一个很好的方法来辨识地址簇，这需要很多背景知识和推测，可能仅仅对特征比较显著的案例有效。

利用交易进行标记

如果仅仅是通过访问交易所或者商家的网站，以查询其公布的接收比特币的地址，会怎么样呢？这其实没有实际意义。因为大多数服务提供商都会针对每一个交易公布一个新的地址，而这个新地址还没有公布在区块链网络上，等待这些地址发生交易没有意义，因为这些地址通常不会再显示给其他人。

唯一可靠的推断地址的方法，是通过和这些服务提供商发生一个实际的交易，交存比特币或者购买一个商品等。当你发送或者接收比特币的时候，你将会知道它们所拥有的地址之一，而且很快这个地址就会在区块链网络上公示（并且是在其中一个簇中的）。于是你可以为这个簇打上该服务商的身份标识标签。

这就是当时“一把比特币”的研究者（以及自那之后的其他人）追踪地址的做法，他们购买了不同的东西，加入了矿池，使用比特币交易所、钱包服务、博彩网站，以及其他一些和这些服务提供商产生比特币交易的行为，总计进行了 344 笔交易。

在图 6.5 中，我们又一次看到了图 6.4 的簇，只不过这一次贴上了附加的标签，我们有关门头沟公司和中本聪之骰的猜测是准确的，这些研究者同时辨识出一批其他的服务提供商，而如果不用交易的方式是很难标识它们的。

辨识个人

下一个问题是：我们是否可以对个人做同样的动作？也就是说，我们是否可以关联一些小的簇以辨识个人在真实生活中的身份？

直接交易。任何人和某个个人进行比特币交易的时候——不管是线上还是线下的商家、交易所，或者一个用比特币来分担晚餐账单的朋友——都可以通过这种直接交易，了解到他们的有效地址（至少一个）。

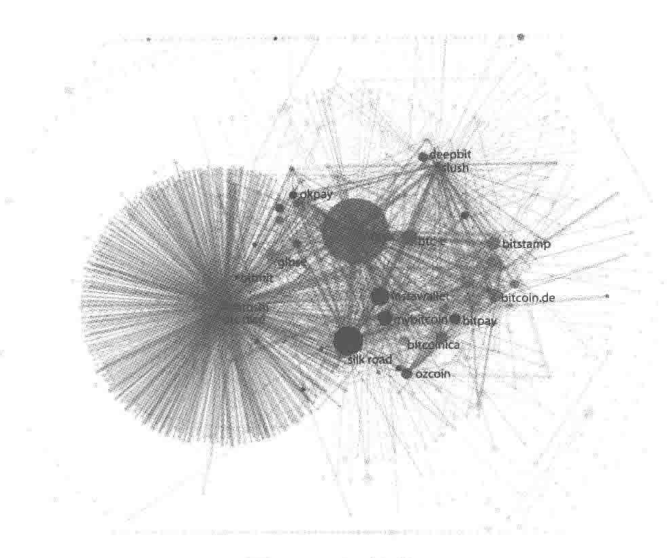


图 6.5　标签簇

注：通过和不同的比特币服务提供商进行交易，米克尔·约翰等人得以辨识并且标记这些簇在真实世界中的身份。

通过服务提供商。在使用比特币几个月甚至几年的时间里，大多数用户都会跟交易所或者其他中心化的服务提供商有一些交集，这些服务提供商都会直接询问用户的真实身份——通常法律要求它们必须这样做。这个话题我们将会在下一个章节讨论。如果执法部门想要去辨识某一个个人，就可以直接去找这些服务提供商，要求它们提供数据。

疏忽。人们通常都会在公共论坛里公示自己的比特币地址，一个通常的原因都是通过这种办法请求捐助。当有人这么做的时候，其实已经创建了一个他们自己的身份和他们某一个地址的关联，如果他们不使用我们将要探讨的匿名服务，所有的交易都将会面临被暴露的风险。

随着时间的推移，针对隐私的攻击会变得越来越有效率。历史记录表明，当越来越多的研究者去研究并开发出新的去匿名化的技术时，越来越多的数据会被公开，去匿名化的算法也由此随着时间的推移而不断得到改进。除此之外，会有越来越多的辅助信息可以帮助攻击者去识别这些地址簇，如果你非常关心隐私，那么这个问题就值得去担忧。

目前，我们探讨的去匿名化技术，都是基于对区块链网络上交易图谱进行

的分析，这些方法被归纳为**交易图谱分析**（transaction graph analysis）。

网络层的去匿名化

用户被去匿名化，有很多种不依赖于交易图谱的方法。为了在区块链网络中公示一个交易，一种典型的方法就是广播这个交易到比特币点对点的网络中，在这个网络中，消息会被相应地发送，但不一定要在区块链网络里做永久记录。

在计算机网络术语中，区块链被归为应用层，而点对点的网络则是网络层。2011年，丹·卡明斯基（Dan Kaminsky）在黑帽技术大会（Black Hat）上首次提出了网络层去匿名化的概念。他注意到，当某个节点创建一个交易时，该节点就会和其他很多节点建立链接并且广播该笔交易。如果网络上足够多的节点串通起来（或者是被同一个攻击者所控制的），他们就能分辨出第一个广播交易的节点，并且可以因此推断，这个节点就是被创建这个交易的用户所拥有的。攻击者因此可以把这个交易关联到这个节点的IP地址，而IP地址已经非常接近于真实世界的个人身份了——有很多办法可以发现某个IP地址背后的用户身份。因此，网络层去匿名化就是隐私保护的一个非常严重的问题（参见图6.6）。

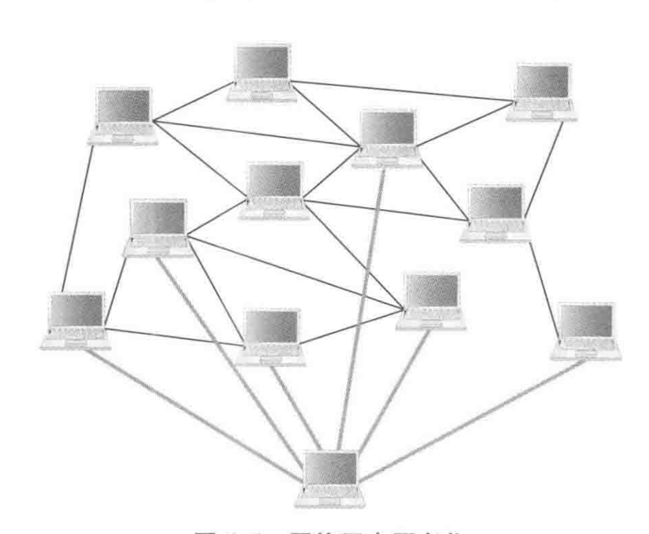


图6.6 网络层去匿名化

注：正如丹·卡明斯基在2011年黑帽技术大会上的演讲中指出的，“第一个通知交易的节点很有可能就是交易源头”。当有多个节点配合并且对同一个交易源头进行识别的时候，这种方法的实际效果会更加明显。

幸运的是，这是一个通信匿名性的问题，已经有很多研究在探索这个课题。正如我们前面在6.1节中已经看到的，Tor这个使用很广泛的系统就是用来实现通信匿名性的。

在使用Tor系统为比特币实现网络层匿名化的解决方案的时候，有几个注意事项。首先，在Tor的协议和任何基于此协议的上层协议之间，可能会有一些复杂的交互，由此可能会导致新的破坏匿名化的方法。事实上，研究者已经发现，在使用Tor协议之上的比特币时，存在一些潜在的安全问题，使用这个方案的时候一定要非常小心。其次，可能有其他一些匿名通信的技术，会更适合比特币的使用。Tor的定位是针对那些低延迟的活动，比如网页浏览。在网页浏览的时候，你也不想坐在那里等半天，因此要取得低延迟，在匿名化方面可能要做出某些牺牲。相反，比特币则是一个高延迟的系统，因为比特币交易需要花时间来获得区块链上的确认。因此，至少在理论上我们可能更希望使用另外一种替代方案来实现匿名性，比如混币网络（Mix Net，参见本章6.3节）。但就目前来说，作为一个实际在运行的并且有广大用户基础的系统，Tor还是有一些优势的，而且这些用户的安全问题已经被集中地研究过。

到目前为止，我们已经看到，通过交易图谱分析的方法，不同的地址有可能被关联在一起，甚至有可能进一步被关联到真实世界的身份。我们也看到，基于点对点网络，交易或者地址可能会被关联到一个IP地址。对后一个问题，虽然我们现在还不能说可以完全解决，但至少解决起来相对容易。前一个问题就要麻烦很多，我们将在本章的后续部分，继续探讨如何去解决它。

6.3 混币

有一些机制可以使得交易图谱分析变得不那么有效，其中一种就是混币（Mixing），这种技术背后的逻辑其实很简单：如果你想要匿名化，那就使用一个中介媒体。这个原则不是特别针对比特币的，在很多需要实现匿名性的情形下都很有用。图6.7展示的就是混币模式。

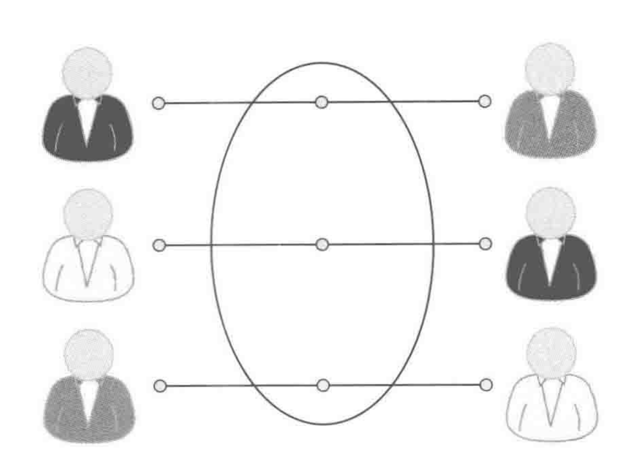


图 6.7　混币模式

注：用户发送比特币给一个中介媒体，并通过其他的用户回收比特币。这就使得在区块链上追踪一个用户的比特币，变得更加困难。

混币在线钱包

如果你还记得我们有关在线钱包的讨论，那么在线钱包貌似就适合作一个交易中介。在线钱包提供了一种在线存储和提取比特币的服务，存储和提取可以在不同的时间发生。通常，你提取的比特币有别于你存储的比特币，那么，是否这就意味着，在线钱包提供了一种有效的混币服务呢？

在线钱包确实提供了一个去关联性的方法，这可以阻止交易图谱分析类型的攻击尝试，在一个具体的案例中，一些杰出的研究者不得不回撤一个面向公众的申明，因为他们之前发现的一个关联，其实只是在线钱包提供的一个伪关联。

从另外一个角度看，使用在线钱包来做混币服务，也存在一些严重的局限性。首先，大多数在线钱包并不能保证一定能实现混币功效，它们实现混币的功能，其实是因为这样做简化了开发工程。作为用户，你得不到它们不会去变更其混币模式的保证。其次，即使它们实现了混币，其内部也会保留一份记录，用来匹配你的存入和提取。这不仅是出于安全性的考虑，也是根据合规要求的审慎做法。所以说，如果你的威胁模型会考虑服务提供商本

身会跟踪你，或者服务提供商被黑客攻击，又或者服务提供商被迫提供它们的记录等这些可能性，你就又回到了原点。最后，除了保留内部记录之外，声誉好的且受监管的服务提供商，也会要求你提供个人身份以进行记录（我们将在第7章中具体讨论监管问题）。你无法简单地通过一个用户名和密码，就能够创建一个账号。所以，在某种意义上，这可能比你不用钱包服务还糟糕。

在线钱包服务所提供的匿名性，和传统银行所提供的服务类似，都有一个知道所有用户的交易记录的中央媒介。对一个没有特权信息的陌生人来说，我们具备了一定程度的隐私。但是，正如我们讨论过的，区块链的公共属性意味着，如果发生任何问题（比如，钱包或者交易所服务被黑客攻击而导致内部数据的暴露），隐私风险会比传统的银行系统更大。除此之外，越来越多的人就是因为不满意传统系统的匿名性，并想要一个更好的（或者不同的）匿名性保障才转向使用比特币，这些都是用户使用专项混币服务的动力。

专项混币服务

不同于在线钱包，专门的混币服务既可以保证不留记录，又不需要你的身份验证。你甚至不需要一个用户名或者其他化名来使用这项服务，只需要发送比特币到混币服务提供的地址，并且告诉交易服务提供商你发送的比特币所需要达到的地址，混币服务提供商就会帮你转过去相同数量的比特币（不是你发送的比特币）。本质上，这是一种互换。

同时，另一个好处是这种专项混币服务承诺不会保留记录，这看上去不错。但你必须要信任它们会信守承诺，并且你还必须相信它们最终会帮你完成转账。不像在线钱包，由于混币服务并不提供一个存储比特币的地方，你需要混合后的比特币尽快回到自己手中。这也就意味着，混币池中将要和你刚刚存入的比特币混在一起的其他比特币总量会非常少，大概就是在你的比特币存入的同时其他人存入的数量。

混合（mix）和洗钱（laundry）

在这本书里，我们使用了一个术语混合，特指一个专项的混币服务，也有些人比较倾向于用另外一个同样意思的术语 mixer。

你可能也遇到了洗钱这个词，我们并不喜欢这个词，因为这个词不必要地关联到了一个道德评价的问题，而实际上我们讨论的是一个纯技术的概念。正如我们所见，为什么你需要保护你在比特币运用中的隐私，并且使用混币服务来保护你的日常隐私，你有很多很好的理由。当然我们也需要了解它的负面作用，但是洗钱这样的描述加重了负面含义，因为这可能带了一种暗示，暗示你的比特币是“脏”的，需要洗干净。

还有另外一个词翻洗（tumbler），这次词的含义不是很清楚，可能会被理解为翻滚式混币的一种行为，或者是指因此而带来的清洗效果（在宝石之类的范畴）。不管怎么说，我们还是坚持使用“mix”这样一个词。

混币准则

有一组研究者，包括本书 5 个作者中的 4 个，研究了混币模式，不仅仅是从增强匿名性的角度，而且还从安全信任等级方面，提出了一系列改进混币运作的方案，我们将深入探讨这些准则。

多重混币的使用

首要的原则是使用多重混币服务，一环套一环，而不是仅使用单次混币（参见图 6.8）。这是一个已经被广泛接受并且已经比较完善的原则，例如，正如我们简要探讨过的 Tor 系统，使用了三重路由方式的匿名通信。这可以减少你对单一混币服务提供者可信赖性的依赖。只要这一系列中的任何一个混币服务提供者信守承诺并删除了记录，你就有理由相信，没有任何人能够将你的原始输入关联到你最终接收到的输出。

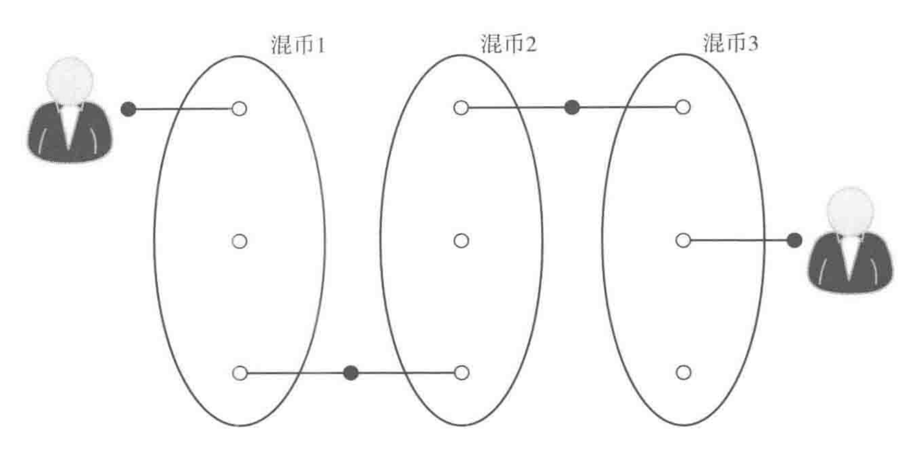


图 6.8　多重混币

注：我们从一个持有比特币或者输入地址的用户开始，且假设该用户已经被其他人关联到了其具体身份。该用户通过混币服务提供商来发送比特币，每次都需要提供一个全新产生的输出地址，只要能够在所提供的至少一个混币环节销毁输入和输出匹配的记录，而且没有其他渠道可以泄露相关信息，其他人就无法把用户最初发送的比特币和最终接收的比特币关联起来。

一致性交易（uniform transactions）

在混币交易中，如果不同用户使用了不同数量的比特币，这样的混币不一定会非常有效。由于初始进入和从混币过程中出来的比特币数量必须要一致，就可以通过观察某用户的比特币在混币过程中的流入流出而建立起一个关联，或者至少会极大地减少匿名集中的交易数量。

相反，我们需要混币中的交易价值能够服从平均分布，从而最小化被关联的可能性。所有的混币服务都应该对使用的块大小（chunk size）达成一致，也就是说，使用一个固定的混币输入值，就可以增强所有通过混币服务的交易匿名性。因为所有的交易看起来都一样，而不能通过交易价值的不同，分辨出不一样的地方。再者，在所有的混币中使用统一的大小，可以让使用多重混币更加容易，而不需要去拆分和合并其中的交易。

在实际运作中，对所有用户统一交易块大小可能会比较困难。如果我们选择的块太大，对想使用混币来处理少量比特币的用户则不适用；而如果选择的块太小，那么想处理大额交易的用户可能需要把交易拆成大量的小额交易，这

种做法会非常没有效率而且成本很高。多标准块大小则可以改善性能，但是不同的块大小也会相应地分割交易匿名集。或许一系列渐增的两到三个块大小，有可能会在效率和隐私程度之间达到一种合理的平衡。

客户端自动化

除了在基于交易量的关联尝试之外，一些聪明的攻击者会尝试其他不同的方法，例如观察交易发生的时间。这些攻击其实可以防范，但必要的预防措施对于人类来说太过复杂和麻烦。相反，混币服务的客户端功能应该是自动化的，并且是隐私保护比较好的钱包内置功能。

手续费应该是要么全有要么全无

混币服务是一种有收益期望的生意。一种计费的方式是从每一笔交易中分成，但是这种方法对匿名性的实施是有问题的，因为混币不再是统一的大小。（如果用户尝试去分拆和合并较小的交易块使得交易大小回到初始的状态，那就有可能带来严重的并且难以分析的匿名被暴露的风险，因为有更多的新的有关交易中的比特币的关联会产生。）

> 不要把交易手续费和混币服务费混为一谈，交易手续费是矿工所获得的，混币服务费是在此之上的额外的费用。

为了避免这个问题，混币服务费应该是要么全有要么全无，并且依概率规则来应用。换句话说，混币服务提供商应该要么在很小的概率情况下获得所有的交易金额，要么完全不收费。举例来说，如果混币服务商想要按照 0.1% 收费，那么应该是每 1 000 次交易中有一次服务提供商获得整个交易金额，而其他的 999 次则不收任何费用。

这个会很难实现，混币服务商需要做出一个概率决策，并且要让用户信服它们没有作弊，也就是说，在它们的随机数生成器中没有做过任何概率偏置设定。比如，获得整个交易金额的概率是 1% 而不是 0.1%。加密学提供了一个很好的办法，你可以参考下面将要延伸阅读章节中提到的有关混币的论文，以获取更多的细节，

在这篇论文中，也提到了可以让混币服务提供商提高公信力的多种其他方式。

混币实践

直至2015年，还不存在一个正常运行的混币生态系统。市场上有很多的混币服务，但是都只有比较低的交易量，因此它们的匿名组合比较小。更糟糕的是，许多混币服务提供商被报告有盗币行为，或许“自举”这样一个生态系统太难，正是混币系统从来没有良好运行过的一个原因。基于混币服务提供商狡猾的名声，并没有多少人想要使用它们的服务，这也导致了较低的交易量，进而导致了不好的匿名保护。老话说得好，大隐隐于市（anonymity loves company），也就是说，越多人使用一个匿名服务，那么这个服务能提供的匿名性就越高。进一步来说，由于提供服务并没有太多利益可图，服务提供商可能会尝试去盗取客户的资金，这会使得混币服务提供商的公信力出现持续的恶性循环。

当前，混币服务提供商并没有遵循我们所探讨的任何原则。每一个服务提供商都是独立运营的，并且通常都会提供给用户一个网页接口，让用户手工输入收钱地址和其他一些必要的参数。用户可以选择他们需要进行混币交易的数量，服务提供商针对每一笔交易收取提成来作为服务费，然后发送剩余的比特币到用户指定的目标地址。

我们认为，对混币服务提供商（和钱包软件提供商）来说，为了可以获取更强的匿名性，抵御智能攻击，提供一个高可用性的接口，进而吸引更多的交易量，实施我们介绍的模式是很有必要的。然而，迄今为止，我们还是没有看到过一个比较强健的混币生态系统。

6.4 分布式混币

分布式混币（Decentralized Mixing），不同于一般的混币交易，指的是用一种用户之间的点对点模式实现混币交易的协议。正如你可以想象的，这种方式在理念上与比特币更加契合。

分布式模式具有更高的可操作性。首先，分布式没有自举的问题，用户不需要等待一个有公信力的集中式混币提供商出现。其次，盗币行为在分布式混币模式下几乎不太可能发生，这种协议可以保证你可以收回你在进行混币交易时等值的比特币，正是因为这一点，即便是要进行一些对分布式混币有用的中心化的协作，由于无须说服别人自己是值得信任的，任何人都可以更加容易地设置并提供这样的服务。最后，在某些方式中，分布式混币模式可以提供更好的匿名性。

合币

分布式混币模式的主要方案被称为**合币**（Coinjoin）。在这个协议中，不同的用户共同创建一个单一的比特币交易，该交易包含所有的用户输入。让合币得以有效运作的核心技术原理为：当一个交易拥有多个来自不同地址的输入时，来自每一个输入的签名都是分离并且相互独立的，所以这些不同的地址可以被不同的人所控制，而不需要任何一方来提取所有的私钥（参见图 6.9）。

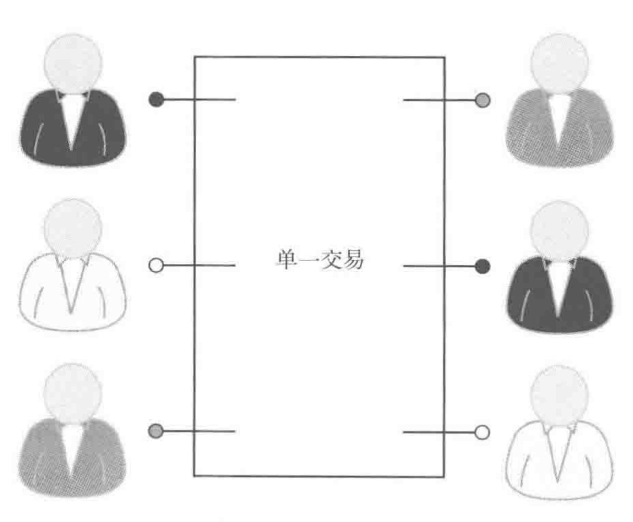


图 6.9　合币交易

这样就可以让一组用户通过使用单一交易来进行混币交易。每一个用户提供一个输入和输出地址，然后组合起来就形成一个交易。由于输入和输出地址

的顺序是随机的，因此攻击者无法建立输入和输出的匹配关系。参与者可以查询他们的输出地址已经包含在交易里，并且会接收到和他们输入时数量相同的比特币（减去所有的交易手续费）。一旦他们确认了这些，就可以对这个交易进行确认签名。

当其他人在区块链网络上查询这笔交易的时候，即使他们知道这是一笔合币交易，也不能确定输入地址和输出地址的匹配。从一个外来者来看，这些比特币已经被充分混币了，这就是合币的精髓。

到目前为止，我们已经描述了一轮混币交易，但是我们在6.3节中讨论的原则仍然适用，你想要（想必会）和不同组的用户重复这样的流程，你也想保证这些交易比特币块大小是标准的，这样就避免了无意中引入会造成信息泄露的旁路。

现在让我们来探究一下合币的细节，我们可以把该流程分为5个步骤：

1. 找到想要混币的交易对手，作为节点。

2. 交换输入/输出地址。

3. 建立交易。

4. 发送这个交易给其他人，每一个节点在确认他们的输出地址之后，进行签名。

5. 广播这个交易。

一组想要进行混币的节点，首先需要发现彼此。这个动作需要由一个服务器来完成，其角色有点像一个“饮水坑”（watering holes）[①]，允许这一组用户互相连接并组合在一起。不像中心化的混币服务，这些服务器既不可能有机会盗取用户的资金，也不可能危及用户的匿名性。

一旦一组节点形成，这些节点必须要互相交换它们的输入和输出地址。地址交换中要保证即便是组中其他节点也不能知道这些输入和输出地址之间的匹配关系。这一点非常重要，否则即使你与一组看上去随机的节点进行了合币交

① 这个比喻来自非洲草原上的饮水坑，需要饮水的动物都会到坑边去。掠食动物也常常在坑边埋伏捕猎。在网络安全上，这种类似“饮水坑”的网站也往往是黑客选择攻击目标的场所，此类攻击也被称作“水坑攻击”。

易，攻击者还是有可能伪装自己进入这个节点组，并由此获取输入和输出的对应匹配。要做到无关联的地址交换，我们就需要一种匿名通信协议。我们可以使用之前探讨过的Tor网络，或者一种被称为加密混币网络（mix-net）的特殊目的匿名路由协议。

输入和输出地址信息一经传达，其中一个用户——不管是谁——将会基于这些相对应的输入和输出地址构建一个交易，这个未被签名过的交易将会被转发传递，每一个节点都会验证这个输入和输出地址是否正确，并且签名确认。

如果所有的节点都遵循这个协议，那么这个系统就会正常工作。任何一个节点都可以组装交易，并且任何一个节点都可以将这个交易广播到网络中，甚至这些节点中的两个可以独立广播，当然，这个交易只会在区块链网络中被公布一次。但是，如果一个或多个节点想要进行破坏，那么启动一个拒绝服务攻击并阻止这个协议完成，是很简单的事情。

特别地，一个节点可以参与协议的第一阶段，提供输入和输出地址，但拒绝在第二阶段进行签名。或者，也可能是，在对交易进行签名确认之后，一个想要破坏交易的节点可以尝试使用它所提供给其他节点的输入地址到其他的交易中去，如果另一交易在网络中被抢先确认，那么合币交易就会被当作双重支付交易而被拒绝。

在合币交易中，有多种方案可以防止拒绝服务式攻击。其中一个就是对协议中参与交易的节点施加成本，不管是通过一种工作证明机制（类似于挖矿），或者是通过一种销毁证明机制（一种可证实能销毁你所拥有的微量比特币的技术，这一点我们在第3章探讨过）。另外的办法包括，使用一些密码学手段鉴别不符合规定的参与者，并且可以把它们从节点组里剔除。在本章结尾处，我们会看到一些相关细节。

高风险交易流（high-level flows）

我们之前谈到过旁路攻击。现在我们来仔细看一下产生旁路的玄机。我们假设，通过一个特定的地址，爱丽丝每周都固定地收到一定数量的比特币，比如43.12312个比特币，有可能这是她的薪水。进一步假设她有一个习惯，每当

收到这笔资金的时候就把其中的5%立刻自动存入另外一个比特币地址，那是她的退休基金账号。我们将这种转账模式称为高风险交易流。在这种情况下，没有一种混币模式可以隐藏这两个地址之间的关系，考虑到这种行为模式会在区块链网络中是透明可见的——这样特定了金额和时间的行为，几乎不可能是偶然发生的。

有一种技术，可以帮助用户在高风险交易流的情形下重获无关联性，这种技术叫作**合并规避**（merge avoidance），是由比特币创始人迈克·赫恩（Mike Hearn）提出的。一般来说，为了完成一笔支付，用户会尽可能地组合所拥有的比特币，以便有足够多的数额可以支付到单一接收地址来完成交易。他们是否可以规避会导致所有输入地址被关联在一起的合并行为呢？这种合并规避协议通过允许接收方提供多个输出地址的方式（尽可能多的），使得无关联性成为可能。发送方和接收方可以达成一致，通过把一个数额较大的支付分拆为一组小面值的支付方式，使得这个支付使用多个交易来完成，如图6.10所示。

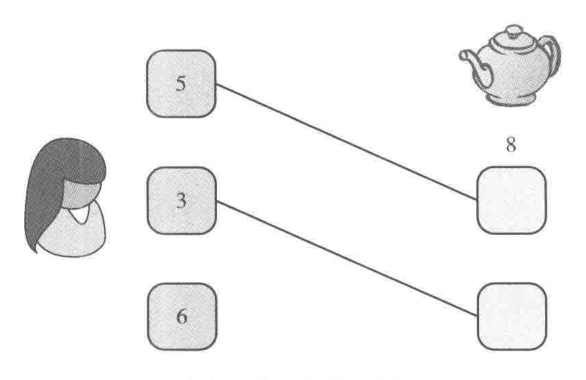


图6.10　合并规避

注：爱丽丝想要用8个比特币去购买一只茶壶，店铺提供了两个地址给她，她可以支付5个比特币到其中一个地址而支付3个比特币到另外一个地址，与她的可用输入资金匹配了，这样就可以避免暴露两个地址都是属于爱丽丝的事实。

假设，店铺最终将这两笔支付和收到的其他支付合并在了一起，这两个地址就不再是很明显互相关联的了。店铺应该避免在收款的同时，马上重新把这两个输入合并在一起，否则这两个输入来自同一个个体的事实还是会很明显。当然，爱丽丝也要避免在同一时间发送这两笔支付交易，这也有可能类似地暴

露信息。

总的来说，合并规避可以帮助缓解高风险交易流的问题：如果某一个交易流水被拆分成许多互相无法关联的小额流水，攻击者可能就不会辨别这个交易。这样做也可以使依赖于识别单一交易中共时花费的多个比特币的地址簇技术失效。

6.5 零币和零钞

在零币和零钞陆续出现之前，还没有哪个加密数字货币的匿名化方案可以让人如此兴奋，不光是因为它们所实施的密码学原理非常高妙，也因为它们承诺可以达到的匿名性非常强大有效。到目前为止，所有我们看到过的匿名加强技术，都是在原来的核心技术协议之上加载匿名化处理，零币和零钞则是在协议层就融合了匿名化处理。我们将在这里以比较大的轮廓阐述一下这个协议，并对某些细节进行了必要的简化，但读者可以在本章结尾的延伸阅读部分找到原始论文作为参考。

兼容性（compatibility）。零币和零钞为保证很强的匿名性，是有代价的：不同于中心化的混币服务和合币交易，这些协议和现在的比特币不兼容。通过软分叉（soft fork）① 比特币协议的方式去实施零币在技术上是可行的，但实际操作的难度会对此造成很大的障碍。零钞甚至不能通过软分叉比特币协议的方式去实现，而只能以一种另类币（altcoin）的方式存在。

加密学保证（cryptographic guarantees）。零币和零钞在协议层已经融入了混币功能，其匿名属性来自加密学的保证，从性质上说，这些保证比我们之前讨论的其他混币技术更好。在隐私保护方面，你不需要信赖任何人，比如，混币服务提供商、混币节点，或其他任何形式的中介，甚至是矿工和共识机制协议。和大多数密码学保证一样，这种匿名性的保证仅仅依赖于攻击者的计算能力上限。

① 所谓软分叉在比特币中的含义是对协议进行向前兼容的修改。修改后的新版本会造成原有的部分区块或交易无效，但是按更新后的协议产生的新交易和区块在旧协议下是有效的。换句话说，新协议是原协议的一个子集。——译者注

零币

为了解释零币，我们首先要介绍一下**基础币**（Basecoin）的概念，基础币是一种类似于比特币的另类币，而零币是这种数字货币的一种延伸，其所提供的匿名性的核心特点在于，你可以将基础币和零币进行来回转换，并且当你这么做的时候，就打破了旧的基础币和新的基础币之间的关联。在系统里，基础币是你需要进行交易的货币，零币只是提供了一种交易基础币的机制，这种机制可以确保新币和旧币之间毫无关联。

你可以把你所拥有的每一个零币当作一个令牌，用来证明你拥有这么一个零币并且使其不能再被消费。这种证明机制并不会显示你所拥有的是哪一个零币，而仅仅是证明你确实拥有一个零币，稍后你可以将这个证明给矿工看，以赎回这个证明并取得一个新的基础币。用一个比喻来说，就好比你去赌场用现金换了一些扑克筹码，这些筹码就是一种证明，证明你存了多少现金，等你离开赌场时，就可以拿着这些证明去换相同数量的但并不一样的现金。当然，不像扑克筹码，除了你可以稍后用来赎回一个基础币，你并不能拿零币做任何事情。

为了在加密数字货币中让这样的机制正常运转，我们要用密码学的方式来执行这些证明，我们需要确保每一个证明只能赎回一个基础币，否则你就可以通过把一个基础币转换成一个零币，然后多次赎回获取更多的免费基础币。

零知识验证

我们使用的核心加密学工具是零知识验证，这种方式可以证明一个声明（数学上的）是正确的，而不需要展示可推导该声明正确性的任何其他信息。例如，假设你已经做了很多工作解决了一个哈希谜题，并且你想要向其他人证明你做到了。换而言之，你想要证明“我做到了”这个声明。

I know x such that H（x ‖ 〈other known inputs〉） < 〈target〉

当然，你可以展示上述公式里的 x 值来证明你做到了，但是零知识验证可以让你向别人证明你做到了这一点，同时不需要透露 x 的值，即便在看过你的证明之后。

你也可以证明一个如“我知道一个 x 值，而公式 H（x）的结果属于下面这一个集合 {…}”这样的声明。该证明既没有展示 x 值是什么，也没有证明集合里面到底哪一个元素等于 H（x）。至关重要的是，零币就是利用零知识验证来实现其功能的。事实上，零币中被这种方式证明的声明，与后面要提到的例子非常相似。本书中，我们把零知识验证当成一个黑匣子，只说明了零知识验证可以实现的属性以及在这个协议中的哪个部分是必需的，我们并没有深究如何实现这一功能的技术细节。零知识验证是现代密码学的一个基石，是很多相关技术协议的基础。再一次强调，我们建议有兴趣的读者可以参考延伸阅读中提到的文献，去了解更多更加详细的内容。

铸造零币

零币通过铸币过程而产生，而且任何人都可以铸造一个标准面值的零币。为简便起见，我们认为零币只有一种面值，每一个零币价值一个基础币。虽然任何人都可以铸造一个零币，但是产生的零币并不自动具备任何价值——你不可能获得免费的钱。只有把零币放到区块链网络上，并且通过消耗一个基础币的方式，它才能具备价值。

为了铸造一个零币，你需要使用加密学承诺。回顾一下我们在第 1 章讲过的内容，承诺方案类似于将一个值放入一个信封，并将信封置于所有人的视野中（见图 6.11）。



图 6.11　一个序列号的承诺

注：密码学承诺好比把一个序列号封装到一个信封里。

铸造零币的过程分为三步：

1. 生成一个序列号S和一个随机密钥r。

2. 计算一个函数Commit（S，r），这是序列号S的承诺。

3. 如图6.12所示，在区块链上发布该承诺，这需要消耗一个基础币，此币不可再被花费，进而创建了一个零币。此时并S和r仍然是保密的。

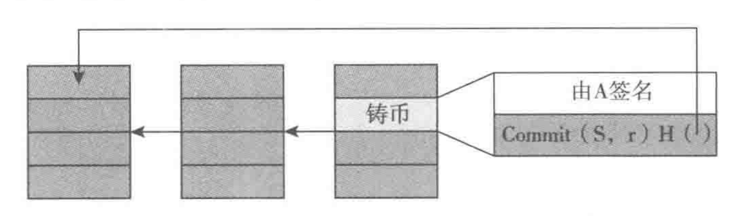


图6.12　在区块链网络上设置一个零币

注：为了将一个零币置于区块链中，需要创建一个铸币交易，其输出地址是零币序列号的一个密码承诺，而铸币交易的输入则是一个基础币，这个基础币也会在创建零币的过程被消耗掉，整个交易过程并不需要公示这个序列号。

为了消耗一个零币并赎回新的基础币，你需要证明你之前已经铸造了一个零币，你可以通过公开之前的承诺也就是说公示S和r的值来证明这一点，但是这样显然就建立了一个你的旧的基础币和新的基础币之间的关联，那么我们怎样才能打破这个关联呢？这个时候就用到零知识验证了，在任何时间节点，区块链网络上都有很多的承诺对象——我们将其命名为c_1，c_2，…，c_n。

以下是消耗一个具有序列号S的零币以赎回一个新基础币的步骤：

- 创建一个特殊的“花费”交易，这个交易包含序列号S和一个具备零知识验证的声明：“我知道在承诺对象（S，r）中的r在以下的集合里：$\{c_1, c_2, ..., c_n\}$”。
- 矿工将会验证你的零知识验证，这将给予你打开区块链中一个零币承诺的能力，而你并不需要真的打开它。
- 矿工也会查询序列号S，确认这个零币没有在之前的花费交易中被使用过（为了防止双重消费）。
- 你的花费交易的输出将形成一个新的零币，你应使用你所拥有的一个地址来作为输出地址。

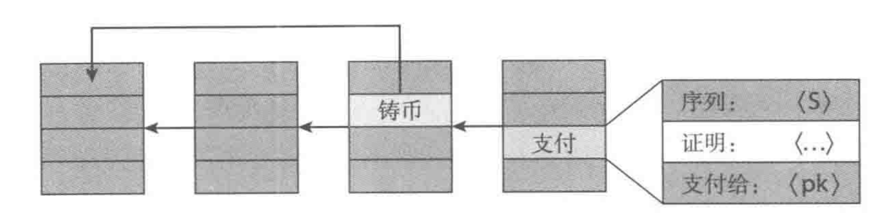


图 6.13　花费一个零币

注：花费交易展示了之前铸币交易中所锁定的序列号 S，以及 S 和之前的铸币交易的关联性的零知识验证。不像一个铸币交易（或者一个普通的比特币/基础币的交易），这个花费交易没有输入地址，因此也没有签名，只有一个零知识验证的证明来确立它的有效性。

一旦你花费了一个零币，其序列号就变得公开了，那么你就不能再次赎回同一个序列号所对应的零币，由于每一个零币都仅有唯一的序列号，正如我们从安全角度所要求的那样，每一个零币只能被花费一次。

匿名性。在整个过程中，我们可以发现 r 一直是保持隐匿状态，不管是铸币交易还是花费交易都没有展示过 r，这意味着没有人知道哪一个序列号对应哪一个具体的零币，这就是零币匿名性背后的核心概念。在区块链上，生成序列号 S 的铸币交易和稍后公示 S 以赎回一个零币的花费交易之间，并没有关联性。这种听起来像魔术般的特性是无法使用实体的封装系统来实现的，但在密码学中是可以做到的。好比我们在桌上放了一些装有不同序列号的密封过的信封，你可以证明某一个序列号是这些信封里面的一个，而不需要展示是哪一个，也不需要打开任何一个信封。

效率。回忆一下我们在花费交易中证明过的一个声明：

“我知道在承诺对象（S，r）中的 r 在以下的集合里：$= \{c_1, c_2, \cdots, c_n\}$”。其中，n 代表的就是曾经被创建过的零币的数量。听起来这种零知识验证的实施会变得非常没有效率，因为证明中包含的集合大小会随着 n 的增加而线性增大。神奇的是，零币可以让这种验证的复杂度仅仅是 n 的对数。我们需要注意到这一点，即使需要被验证的声明的长度是线性的，声明本身并不需要被包括在证明里，而是隐含的。由于矿工们知道区块链上所有零币的集合，声明就可以被矿工们自行推断出来。这样证明本身就可以非常短。尽管如此，跟比特币相比

较，零币还是增加了相当大的额外开销，大概50KB。

建立信任

用于搭建零币的工具之一（RSA 累加器）需要进行一次性信任设置。特别的是，一个被信任方需要选择两个大的质数 p 和 q，并且公示 $N = p \times q$ 作为所有人在系统的整个生命周期中使用的参数。我们可以把 N 看作一个公钥，只不过它被用于所有的零币而不是仅仅对应于某一个体。只要这个被信任方，销毁任何关于 p 和 q 的记录，系统就可以被认为是安全的。需要强调的是，这一做法是基于对两个大质数的积进行因子分解是不可行的这一假设的普遍认可。但是，如果任何人知道了秘密的因数 p 和 q（也被称为“陷阱门”），他们就可以在不被监测到的情形下，为自己创建新的零币。所以，这些秘密的输入只能被使用一次，在用于产生相对应的公钥后就被销毁。

这里有一个有趣的社会学问题，一个个体怎么选择这个 N 并能够让每个人相信，在设置时使用的相对应的质数 p 和 q，已经被安全地销毁了，这一过程还不是很清楚。如何实现这个目标，会有很多不同的方案，其中就包括“阈值加密”技术，该技术可使用多个代理协同计算出 N，而只要其中一个代理将自己的秘密输入销毁，这个系统就可以被认为是安全的。

我们还可以使用其他略有不同的加密构造来避免设置这样一个被信任方。特别是，有实践表明，随机生成一个大数就具有很高的安全性，因为这样一个数字很有可能不能被完全分解。但遗憾的是，这样做会带来很大的效率冲击，因此这种方法并不实用。

零钞

零钞是一种不同的匿名性加密数字货币，它建立在零币的概念之上，但将加密技术提高到了更高的层次。零钞使用的是一种被称为 zk-SNARKS[①] 的密码

① zk-SNARK 的全称是 zero knowledge Succinct Non-interactive ARgument of Knowledge，是近年兴起的一种密码学方法。Snark 在英文中本就有鬼魅的含义，用到这个技术上倒也有几分神似。——译者注

学技术，这种技术可以使得零知识验证更加简洁、更加有效率，要点就在于，系统的整体效率可以达到某一定程度，使得整个网络可以不需要依赖一种基础币而运行，所有的交易都可以以零知识验证的方式进行。如我们所看到的，零币也支持那些本不需要无关联性的普通交易，只不过在其上做了计算量昂贵的混币交易进行补充。这种混币交易由固定面值的数字币组成，交易价值的拆分与合并只能在基础币系统里实现。在零钞系统中，这种差异就不存在了，交易金额的大小被封装在一个承诺中，在区块链上不再可见，密码学证据确保了拆分与合并的正确性，用户并不能凭空创造出零钞。

账本公开记录的唯一内容就是交易的存在性，以及矿工们用来验证系统正常运行所需要的关键属性的证明。区块链网络上既不显示交易地址，也不显示交易价值。唯一需要知道交易金额的用户，是本次交易的发送方和接收方，矿工们是不需要知道的。当然，如果其中存在交易费用，矿工们则需要知道的仅仅是手续费，这点也不会影响匿名保护。

就匿名性和隐私性来说，零钞这种完全不可追踪的交易系统自成门派。因为公开账本并不包含交易金额，零钞对针对混币服务的旁路攻击是免疫的。

建立零钞系统

按照技术属性来说，零钞看起来好得有点不真实。其实它确实也有自己的命门。就像零币，零钞也需要一个“公开参数”来设置这个零知识验证系统。但是不同于只需要一个几百个字节长度的数字 N 的零币，零钞需要的是一个很大的公开参数集——其大小超过 1G 字节。要再次强调的是，为了生成这些公开参数，零钞需要一组随机并且秘密的输入。如果任何人知道了这些秘密输入，就会产生无法监测的双重消费问题，从而危及整个系统的安全性。

在这里，我们不会过多地深入探讨设置一个 zk-SNARK 系统所面临的挑战，这个问题也是一个比较活跃的研究方向，但是截至 2015 年，我们并不知道如何在实际操作中以足够稳妥的方式建立这个系统。迄今为止，zk-SNARK 还没有被实际运用。

综合比较，融会贯通

现在，让我们从匿名性以及实际的可操作性两个方面，来比较以下我们所探讨的这些方案，见表6.1。

表6.1　本章所讨论的匿名技术的比较

系统	类型	对匿名性的攻击	可部署性
比特币	化名	交易图谱分析	默认系统
人工混币	混币	交易图谱分析，恶意的混币服务提供商，恶意的混币对手方	已被应用
多重混币/合币	混币	旁路攻击，恶意的混币服务提供商，恶意的混币对手方	兼容比特币
零币	加密混币	旁路攻击（存在可能性）	另类币，需建立信任方
零钞	不可被跟踪	未知	另类币，需建立信任方

我们是从比特币，这个已经成功部署了的“默认”系统开始的。但比特币只是化名系统，我们看到，强大的交易图谱分析是可能攻击比特币的匿名性的一种可行办法。我们也探讨了聚集大量地址簇方式，以及如何关联真实世界的身份到这些地址簇的方法。

匿名化技术的下一等级，是用人工的方式实现一个单一混币交易，或者通过人工找交易对手的方式来实现合币交易，这会使输入地址和输出地址之间的关联变得模糊，但同时也会在交易图谱中留下了太多的线索。除此之外，混币服务提供商和参与节点也可能是恶意的，或者因为被黑客攻击的，或者被胁迫公布记录的。即使从匿名性来说离完美还很遥远，但混币服务在现实中存在，并且是现今一个可用的选项。

我们讨论的第三个等级的匿名性，是混币服务链或者合币交易。这种匿名性上的改进，来自更少的对于混币服务提供商或者节点组的依赖。诸如标准化的交易区块大小和客户端的自动化等特性，可以最小化信息泄露的可能性，但是还是有一些旁路风险可能会存在。同时，也还是存在一些攻击者可能控制或

者勾结多个混币服务商与参与节点而带来的风险。实现带有混币服务链功能的钱包和服务在技术上是可行的并且应该会被用户所接受，但是据我们所知，一个安全的混币链解决方案还不存在。

接下来，我们探讨了把加密技术直接应用到协议层并提供匿名化的数学保证的零币。我们认为零币的旁路攻击风险还是存在的，但是明显已经优于其他的混币解决方案。不过，零币需要作为相对于比特币的一种另类币的方式发行。

最后，我们探讨了零钞。通过效率上的改进，零钞可以作为一个完全无法追踪——不仅仅是匿名化——的加密数字货币。然而，就像零币一样，零钞和比特币并不兼容，更糟糕的是，零钞需要一个非常复杂的建立流程，数字币领域仍然在研究如何用最好的方式来实现它。

在本章中，我们讨论了很多技术。现在让我们退一步看，比特币的匿名性（或者匿名潜力）是强大的，当和其他一些技术配合的时候会更加强大，尤其是在匿名通信方面。如我们在第 7 章中将要讨论的，这是一些匿名在线市场所使用的强有力的技术组合方式。

尽管匿名化技术是强大的，但同时也是脆弱的。一个错误就可能造成一个我们不希望看到的，但又是不可逆的关联。匿名化有一些显而易见的有害应用，但同时也存在很多有益的应用，所以是值得保护的。虽然道德层面上的区分很重要，我们还是无法在技术层面清楚地辨识。匿名技术看起来具有深入的和固有的道德模糊性，作为人类社会的一员，我们必须学会怎么面对这个现实。

和关于比特币的道德争论一样，比特币的匿名性也是一个很活跃的技术创新领域。我们仍然不知道比特币的哪一种匿名系统，如果存在的话，将会脱颖而出成为主流。这也将是每一个人的机会——不管你是一个开发者、一个政策制定者，还是一个普通用户——每个人都可以参与其中并做出贡献，希望你在本章中里所学到的内容，可以为你提供一些正确的背景知识去采取行动。

延伸阅读

与前面几章中讨论的主题相比，匿名技术在更快地持续发展中，并且是加密数字货币研究领域中更活跃的一个课题。想要跟上这个领域中最新的进展，可以阅读以下列举的论文，以及引用其他论文。

关于交易图谱分析的《一簇比特币》：

Meiklejohn，Sarah，Marjori Pomarole，Grant Jordan，Kirill Levchenko，Damon McCoy，Geoffrey M. Voelker，and Stefan Savage. “A Fistful of Bitcoins：Characterizing Payments among Men with no Names.” In *Proceedings of the 2013 conference on Internet measurement*，New York：ACM，2013.

关于我们讨论的混币技术和有效混币原则的来源的研究：

Bonneau，Joseph，Arvind Narayanan，Andrew Miller，Jeremy Clark，Joshua A. Kroll，and Edward W. Felten. “Mixcoin：Anonymity for Bitcoin with Accountable Mixes.” In *Financial Cryptography and Data Security*. Berlin：Springer，2014.

混币服务实践研究，其中很多种并没有很好的声誉：

Möser Malte，Rainer Böhme，and Dominic Breuker. “An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem.” In *2013 eCrime Researchers Summit*. Washington，DC：IEEE，2013.

比特币论坛里，比特币核心开发者格雷·麦克斯韦（Greg Maxwell）发布的有关合币的内容：

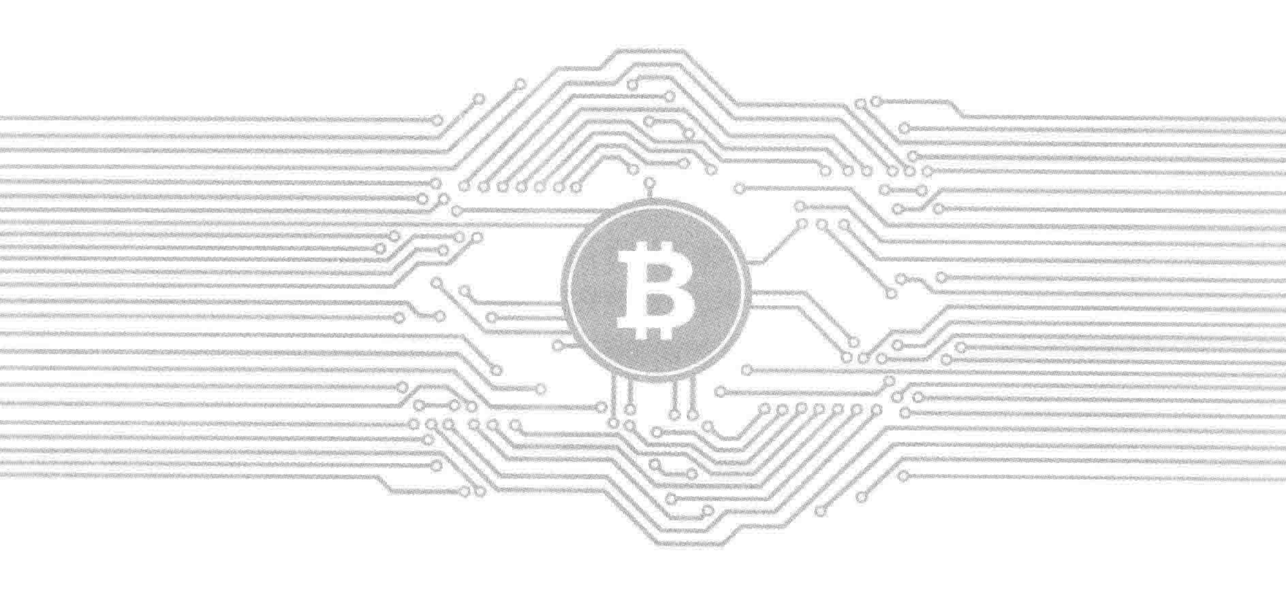
Maxwell，Gregory. “CoinJoin：Bitcoin Privacy for the Real World.” Bitcoin Forum，2013. 下载地址 https：//bitcointalk. org/index. php? topic =279249. 0.

来自约翰·霍普金斯大学（Johns Hopkins University）的密码学者开发了零币，请记住零币和零钞是这本书里我们讨论过的最复杂的加密技术：

Miers，Ian，Christina Garman，Matthew Green，and Aviel D. Rubin. “Zerocoin：Anonymous Distributed E-Cash from Bitcoin.” In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*. Washington，DC：IEEE，2013.

零币的作者和其他一些开发了 SNARK 技术的研究者共同开发了零钞系统：

Ben Sasson，Eli，Alessandro Chiesa，Christina Garman，Matthew Green，Ian Miers，Eran Tromer，and Madars Virza. “Zerocash：Decentralized Anonymous Payments from Bitcoin.” In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*. Washington，DC：IEEE，2013.



第7章

社区、政治和监管

BITCOIN
AND
CRYPTOCURRENCY
TECHNOLOGIES
A Comprehensive Introduction

在本章中，我们将研究比特币世界和数字加密货币技术是如何影响世界的，并探讨比特币社区的内部政治以及比特币是如何与传统政治，即执法和监管问题互相影响的。

7.1 关于比特币的共识

首先，让我们看一下在比特币问题上已达成的共识，它是比特币运行的基础。为使比特币顺畅运行，人们必须就以下三个问题达成共识：

1. 关于规则的共识。这里所说的规则是指包括确保交易或区块有效的机制，及比特币运行时涉及的核心协议和数据格式等内容。人们需要就这些规则达成共识，这样，比特币系统中的所有参与者才能就发生的情况相互沟通并达成协议。

2. 关于历史记录的共识。也就是说，参与者必须对区块链的内容，包括哪些是属于区块链，哪些是不属于区块链的内容达成共识，这样，人们才能就如何确认已发生的交易达成共识。在此基础上，人们就可以对比特币、未动用产出的数额及拥有人达成共识。这一共识源自区块链的创建过程和使各个节点对区块链内容的理解达成一致的过程，我们已经在第 1 章和第 2 章中对这些过程进行了描述。这是比特币中最常用且技术上最复杂的一种共识。

3. 关于比特币价值的共识。第三种共识要求人们普遍认为比特币是有价值

的，比如，如果有人今天给了你一个比特币，你明天就能够将它兑现或用它换取到有价值的东西。任何一种货币，不管是像美元这样的法定货币，还是像比特币一样的数字加密货币，赖以存在的基础都是其具有价值的共识。也就是说，人们普遍接受可以用它进行交易，在现在或未来可以用它换取其他有价值的东西。

对于法定货币，第三种共识是唯一的共识。“货币有价值”这一共识不是由规则决定的，法律规定了它是不是钞票。历史记录并不重要，但是状态很重要——谁拥有什么。状态由物质占有（如持有现金）或专业记录（如银行）来决定。然而，对于数字加密货币，人们还需要对规则和历史记录达成共识。

对于比特币，与其他共识不同的是，这第三种共识具有一定的循环性。即，我相信我今天收到的比特币是有价值的，这取决于我希望明天收到这个比特币的人同样相信它的价值。因此，对价值共识的基础在于对价值延续性的共识。这有时被称为“仙子效应”（Tinkerbell effect），这个名字来源于童话故事《彼得潘》，仙子之所以存在，是因为你相信她存在。

不论是否循环，对于价值的共识都是存在的，这对比特币系统的运行至关重要。而且，还有很重要的一点是，这三种共识相互关联，如图 7.1 所示。

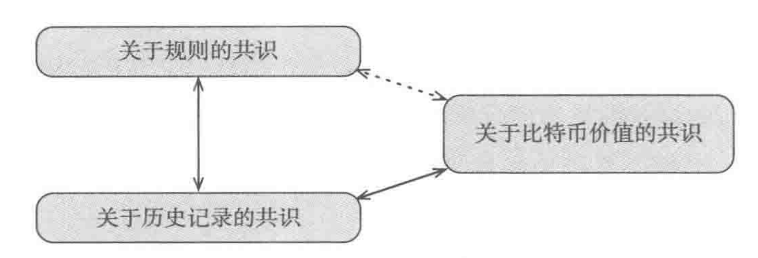


图 7.1　关于比特币的三种共识之间的关系

首先，对规则与对历史记录的共识相互依赖。如果不知道哪些区块是有效的，也就无法对区块链达成共识。如果不能对区块链中有哪些区块达成共识，也就无法判断交易是否有效，进而无法判断有没有双重支付的企图。

对历史记录和对比特币价值的共识也紧密相关。对历史记录的共识意味着我们同意谁拥有哪些比特币，这是比特币具有价值的先决条件——例如我拥有

一个比特币，如果不能通过历史记录对此达成共识，我就不能指望将来某一天我会把这个比特币付给某人换取其他东西。反之亦然——在第2章我们讨论过，对比特币具有价值的共识，激励着矿工维护区块链的安全，这又促使我们对历史记录达成共识。

比特币原始设计的天才之处就在于，它意识到靠自己本身很难达成这三种共识的任何一种。在一个没有身份概念的、去中心化、全世界范围内运行的系统中，要达成关于规则的共识是不可能的。

类似地，对历史记录的共识是一个复杂的分散式数据结构问题，很难靠自己解决。此外，对某种数字加密货币具有价值的共识也很难达成。但比特币的设计以及运行模式表明，尽管无法靠系统本身达成这三种共识中的任意一种，不过可以通过某种方式将这三种共识组合在一起，并让它们以一种相互依存的方式发挥作用。因此，在讨论比特币社区的运作模式时，我们必须牢记，比特币系统的运行取决于参与者的共识，而且这种共识是十分脆弱的，交织着各种技术和社交元素。

7.2 比特币核心钱包软件

比特币核心钱包（bitcoin core）是一款开源软件，是对比特币规则进行讨论和争议的焦点。这款软件由极为宽松的开源（open source）许可证——MIT许可证认证。只要注明版权声明和许可声明，就可以将该软件用于各种用途。比特币核心钱包是目前运用最为广泛的一款比特币软件。即便不利用它进行软件开发，许多人也会通过研究它来了解比特币的规则。在构建其他比特币软件时，人们会借鉴其规则定义部分的内容，包括判定交易和区块的有效性。

> 比特币核心钱包实际是比特币的规则手册。通过研究比特币核心钱包及其相关解释，可以了解到在比特币系统中真正有效的内容。

比特币改进方案

任何人都可以通过“提交请求”（pull requests）按钮，帮助比特币核心钱包进行技术改进，这一过程在开源软件（open-source software）世界极为常见。若想对软件进行更大的改动，特别是对协议进行修改，则可以通过一个较为正式的叫作比特币改进方案（Bitcoin Improvement Proposal，简称 BIP）的流程来实现。因此，如果你有意通过技术改变来改进比特币，你可以把你的想法写下来，根据比特币改进方案的要求，与其他文件一起公开发表。这会触发比特币社区就你的方案进行讨论，并决定下一步行动。虽然任何人都可以提交正式方案，但正如所有开源项目（open-source project）一样，这存在学习曲线。

BIP 以编号序列形式发布，每项方案有一名拥护者，负责宣传方案、协调讨论活动并努力促成方案在比特币社区向前顺利开展或实施。

我们上面所说的内容适用于对技术更改的方案。事实上，也存在一些 BIP，或者只是为了提供信息，传播关于比特币的知识；或者将之前仅在源代码中明确的部分代码进行标准化。而其他一些 BIP 侧重流程，讨论比特币社区如何决策事项。

总之，除了包含规则手册中的内容外，BIP 还包含方案、制定和讨论规则变更的流程。

比特币核心钱包开发人员

要了解比特币核心钱包的作用，我们需要了解比特币核心钱包开发人员所发挥的作用。原始代码的作者是中本聪（Satoshi Nakamoto），我们在 7.4 节还会介绍。现在，中本聪本人已经不再活跃，但还有一群开发人员在维护着软件。有数百名开发者在为这个项目写代码，但只有少数几个人拥有对核心钱包数据库的“调配”（commit）权限。这些核心钱包的首席开发人员持续维护该软件，并决定哪些新代码可以加入软件新版本中。

这些人的权力有多大？从某种意义上说，他们的权力是很大的，因为他们对代码做出的规则改变终将呈现在比特币核心钱包中，这些规则会默认被遵守。

这些人写下比特币事实上的规则手册。但从另一个角度来看，他们根本就没有什么权力。因为这是一款开源软件，任何人都可以复制、修改它（随时创建一个比特币分叉）。因此，如果首席开发人员的表现不被社区接受，社区可能走向不同的方向。

可以这么想，首席开发人员就像在引领游行队伍前进。他们在队伍的最前面，当他们拐弯时，队伍一般会跟着他们拐弯。但是如果他们试图把队伍带入灾难性的境地，那么队伍中的其他成员可能会选择不同的方向。这些首席开发人员可以敦促社区，但是，如果他们试图把系统带入不被社区接受的技术方向，他们并没有正式的权力，来迫使人们跟随他们。

现在来思考一下，作为系统的使用者，如果你不喜欢它的规则或系统运行的方式，能够做些什么，并与集中式货币（如法定货币）进行比较。在集中式货币系统中，如果有异议，你有权退出，也就是说，你可以不用它。你必须想办法把持有的货币卖出，然后移居到使用另一种货币的地方。有了集中式货币，退出是你的唯一选择。

在比特币系统，你当然也有权退出，但是，因为它作为开源系统（open-source system）运行，你就有了对规则进行分叉的权利。也就是说，你、你的朋友和同事可以选择运行一套不同的规则，而且，通过对规则进行分叉，走向与首席开发人员不同的方向。与退出相比，分叉赋予用户更多的权力，像比特币这样开源系统的社区比完全集中系统的社区拥有更多权力。所以，虽然首席开发人员看似一个拥有控制权的集权式实体，事实上，他们并不拥有一个完全集权式管理人员或软件所有者所拥有的权力。

规则分叉

创建软件分支或规则分叉的一种方式是，以新的创世区块创建新的区块链。人们经常通过这种方式来创建另类币，我们将在第10章谈到这个问题。现在，我们来谈谈对规则的另一种分叉，这种分叉不仅对规则进行分叉，还对区块链进行分叉。

在第3章中，我们谈到了硬分叉和软分叉之间的区别，这里我们谈的是硬

分叉。当对规则有分歧时，区块链中会有分叉，导致两个分支。其中一个分支在规则 A 下有效，而在规则 B 下无效，反之亦然。矿工一旦在两种不同的规则下操作，他们就无法合并到一起，因为每个分支都将包含在另一规则下无效的交易或区块。见图 7.2。

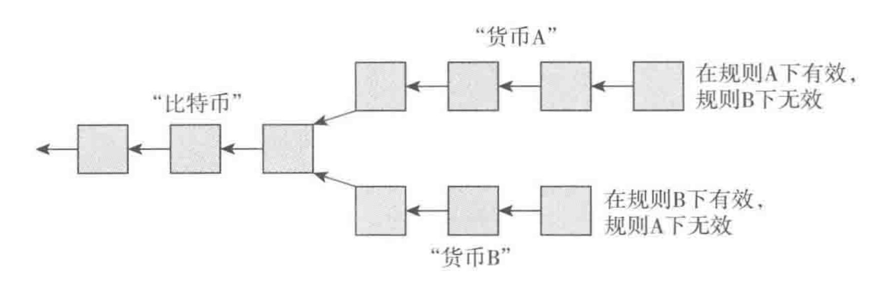


图 7.2　货币中的分叉

注：如果规则的分叉导致区块链的硬分叉，那么货币本身就会分叉成两种新货币。

我们可以把分叉前的货币看作比特币，深受人们认同和喜爱的比特币。分叉之后，出现两种新货币，符合规则 A 的货币 A 和符合规则 B 的货币 B。分叉时，每一个持有一个比特币的人得到一个货币 A 和一个货币 B。分叉后，货币 A 和货币 B 开始分开运行，而且它们可能会独立运行。这两者的规则还可能以不同方式继续发展。

需要强调的是，不仅仅是软件，或规则，或实现规则的软件分叉了，货币本身也分叉了。这个有趣的现象只可能发生加密数字货币中，却不能发生在传统货币中，因为传统货币并不允许用户将货币进行分叉。就目前我们所了解的情况，不管是比特币还是任何另类币，尚未以这种方式分叉过，但确实存在这个奇妙的可能。

人们会对这样的分叉做出何种反应呢？这取决于分叉的原因到底是什么。第一种情况是，进行分叉的原因并不是对规则存在异议，而是想创建一种另类币。如果想创建一种与比特币规则类似的另类币，有些人会通过将比特币的区块链分叉的方式来实现。这对比特币社区来说并不构成真正的问题，另类币单独运行，与比特币主分支和平共存，一些人偏好比特币，另一些人更偏好另类币。但是，正如之前我们说过的，截至目前，还没有人通过将比特币或现有另

类币的区块链分叉的方式来创建新的另类币，他们一般都通过新的创世区块来创建。

一个有趣的情况是，如果分叉的原因是人们对比特币的未来发展存在分歧，换言之，比特币社区内部发生叛乱，因为一些成员认为自己对系统如何运行有了更好的想法，决定脱离出去。在这种情况下，两个分支成为对手，会争夺市场份额。货币 A 和货币 B 都会努力说服更多商家接受它，让更多人购买它。每种货币都想成为“真正的比特币”。它们都声称自己是合法的，并将对方描述成一个怪胎，这可能会引发一场公关之战。

结果很可能就是某一分支胜利，另一分支则渐渐消失在人们的视线中。这种类型的竞争往往指向某一方向。一旦人们认为这两个分支中的某一个更合法并获得了更大的市场份额，网络效应会越来越明显，而另一种货币将成为一种利基（niche）① 货币并将最终消失。获胜方的规则和管理架构将成为比特币事实上的规则和管理架构。

7.3　利益相关者：谁是掌权者

比特币的利益相关者有哪些？真正掌权的又是谁？我们已经讨论过比特币的基础是人们对它达成的共识以及它的规则手册是如何编写而成的，分析过分叉的可能性以及因为对规则存在异议而可能引发的斗争，那么，谁有权决定斗争的胜利者呢？现在我们来谈谈这个问题。

换言之，如果比特币社区存在关于规则设定的讨论和谈判，而谈判失败了，我们想知道谁将对最终结果有决定权。通常来说，在一场谈判中，对谈判协议拥有最佳选择的一方具有优势。因此，搞清楚谁更有可能获胜，将帮助我们了解谁会在关于比特币未来的讨论和谈判中占据上风。

---

① （商业用语）是指针对企业的优势细分出来的市场，这个市场不大，而且没有得到令人满意的服务。产品推进这个市场，有盈利的基础。在这里特指针对性、专业性很强的货币。——译者注

我们代表许多不同的利益相关者做出如下声明：

1. 比特币核心钱包首席开发人员拥有权力——他们编写规则手册，几乎人人都要使用他们的代码。

2. 矿工拥有权力——他们编写历史记录，决定哪些交易是有效的。如果矿工选择某一套特定的规则，理论上来说，其他人也必须要遵守它。拥有更强采矿能力的分叉将创建一个更强大、更安全的区块链，因此，更有能力把规则朝某一方向推进。矿工到底拥有有多大的权力取决于分叉是硬分叉还是软分叉，但不管是哪种分叉，他们都是拥有一定权力的。

3. 投资人拥有权力——他们购买并持有比特币，决定了比特币是否具有价值。可以说，开发人员决定了对规则的共识，矿工决定了对历史记录的共识，那么，投资人决定了对比特币价值的共识。在硬分叉的情况下，如果大多数投资人决定把他们的钱投入货币 A 或货币 B 中，那么该特定分支就被认为是合法的。

4. 商家及其客户拥有权力——他们构成对比特币的主要需求。我们在第 4 章中讨论过，虽然投资人能够一定程度上支持货币价格，但推高货币价格的主要需求来源于将比特币作为一种支付技术促成交易的需求。因此，投资人只是对未来比特币的需求做出推测。

5. 支付服务商拥有权力——它们处理交易。许多商家并不在意它们使用的是哪种货币，它们只想与一家支付服务商合作，支付服务商允许顾客用加密数字货币进行支付，承担全部风险，并在每日结束时跟自己结账。因此，很有可能是支付服务商构成了主要需求，商家、顾客和投资人只是跟随者。

你可能已经猜到了，这些观点都有其合理性，所有这些群体都有一定的权力。要想成功，一种电子货币需要以下不同形式的共识——由开发人员编写的稳定的规则手册、采矿能力、投资、商家和顾客的参与以及支持他们的支付服务商。所以，所有这些参与方都对影响比特币未来发展的斗争结果有一定的话语权，但没有哪一方是拥有绝对控制权的。这是一个庞大、曲折且混乱的建立共识的过程。

另一个与比特币管理相关的组织叫比特币基金会（the Bitcoin Foundation），

它成立于2012年，成立之初是一家非营利组织。现在它主要扮演两个角色：其一，它从资产中拿出一部分资助比特币核心钱包开发人员，以便他们可以全力以赴开发软件；其二，它与政府，特别是与美国政府沟通，作为比特币的发声机构。

现在，比特币社区的一些成员认为，比特币的运行应该游离并独立于传统的国家政府。他们认为比特币应该跨国运行，它不需要向政府解释或证明自己，也不需要与他们谈判。其他人则不这么认为。他们认为被监管无法规避，甚至是有益的。他们希望政府了解比特币社区的利益所在，听到比特币社区的声音。比特币基金会的诞生部分是为了满足这个需求，可以说，比特币被人们理解和接受，很大程度上要归功于比特币基金会一直以来与政府之间的沟通工作。

开放协议（open protocol）的治理

我们已经对比特币系统进了一些描述，在这个系统中，利益并不完全一致的众多利益相关者在开放协议和系统中相互协作，达成技术上的和社会性的共识。这可能让你想起互联网本身。比特币核心钱包和互联网的发展过程确有共同之处。例如，BIP就类似于“评议请求”（Request for Comments，简称RFC），RFC是一种用于设置互联网标准的文件。

比特币基金会一直备受争议。基金会的一些董事会成员卷入了犯罪或金钱丑闻，人们对他们在多大程度上能代表比特币社区存在疑问。基金会面临着要迅速调整此类将带来负面影响的董事会成员，但这会带来挑战。人们指责它缺乏透明度，而且正在迅速走向破产。截至2015年，比特币基金会能够在比特币的未来发展中发挥多大的作用尚不明确。

另一个非营利性组织“货币中心”（Coin Center）成立于2014年9月，总部位于华盛顿特区，承担了比特币基金会的部分职能，充当宣传和与政府沟通的角色。货币中心的运作类似一个智囊团，截至2015年年初，它受到的争议较

小。比特币基金会和货币中心对比特币的控制权都比不上其他的任何一方利益相关者。同开源代码的生态系统中的所有事物一样，这种代表性实体机构是否能成功、其合法性能否获得公众认可，取决于随着时间的推移，它们能够在比特币社区获得多少支持和资金。

总之，还没有一个实体机构或群体对比特币的演化拥有绝对的控制权。从另一种意义上来说，每个人都能够决定比特币未来的演化，因为管理比特币的正是人们对比特币系统如何运作所达成的共识——对规则、历史记录和价值这三个相互关联的方面所达成的共识。任何一种规则集合、群体或管理架构，只要能够维持在这三方面的持久共识，就能够在真正意义上决定比特币的未来。

7.4 比特币的起源

现在我们来谈一谈比特币的起源。它是如何开始的？它的前身是什么？我们对其神秘的创始人了解多少？

密码朋克和数字货币

比特币其中一个值得一提的前身是密码朋克（cypherpunk），一项汇聚了两种观点的运动。首先是自由主义，特别是认为如果没有或者极少政府干预会让社会更好的观点。其次，该运动与自由主义者（甚至是无政府主义者）的概念联系在一起，加上强加密的想法，特别是于 20 世纪 70 年代后期出现的公钥密码学。参与密码朋克运动的人们相信，拥有了强大的网络隐私和强加密，他们可以重塑人们相互交流的方式。密码朋克认为，在这个世界上，人们应该更加有效地保护自己和自身权益，少受政府行为影响（或干预）。

密码朋克遇到的一个难题是，在他们所构想的未来世界中，人们通过强大的技术和加密手段进行网上沟通时，如何解决金钱相关的问题。许多研究都是为了探讨这个问题，特别是大卫·乔姆（David Chaum）和其他人所做的关于数

字货币（digital cash）的早期研究工作，他们试图创建一种具有现金的功能、能够匿名使用和极易交易的新型数字货币。这些技术性的想法是如何发展、数字货币又为何没能流行开来，这背后是一个有趣的故事（参见前言部分）。该领域的所有前期工作都与密码朋克的信仰密不可分，特别是他们对拥有去中心化的、线上和相对私密的强势货币的愿望，都为比特币的诞生播下了种子。它也是很多比特币追随者所遵循的理念基础。

中本聪

2008年，中本聪发表题为“比特币：一种点对点的电子现金系统”的白皮书，宣告了比特币的诞生。白皮书当时可以在网络上自由下载，是第一篇描述比特币的运作模式和设计理念的文章。现在，它依然可以帮助人们迅速了解比特币的技术设计和理论基础。中本聪随后发布了实现白皮书中规范的开源软件，这正是一切的开端。时至今日，中本聪的身份依然是关于比特币的最大谜团。

重要的是，我们不知道中本聪的身份并不是什么要紧事，因为比特币的显著特点就是它的去中心化，而且不受任何单一实体控制。中本聪并不是掌权人，事实上，自从2010年年中将比特币的控制权转让给其他开发者之后，他便不再积极从事这个项目了。从某种程度上来说，中本聪到底在想什么已经不重要了。如果中本聪再度活跃起来，他在比特币社区的声望可能会影响社区的决策，这是他唯一的影响力。

增长

自2009年1月正式上线以来，比特币已经取得了大幅度增长。我们从一段时期的交易走势（见图7.3）和一段时期的交易数量（见图7.4）中都可以很清楚地看到这一点，尽管有过一段时期的滑落，但2013年下半年又开始企稳，并于2015年达到峰值。虽然有时候，增长是渐进的，但也有激增的情况，通常是在新闻事件发生之后。总的来说，长期看增速是加快的。

谁是中本聪?

几乎可以肯定“中本聪”是一个笔名。中本聪自称是一名生活在日本的37岁中年男性。但是，目前没有证据表明中本聪说日语或懂日语，但是他的英文书写相当流利，尽管美式拼写和英式拼写混在一起。有不少人研究中本聪的文章、代码、发表时间、机器标识符等蛛丝马迹，试图以此回答中本聪的母语是什么，他来自哪里。有些人甚至尝试通过文体计算（对作家风格进行文字算法分析）来找出中本聪的身份。虽然有些人自称是中本聪，甚至一家新闻媒体也曾这样宣称过，目前，中本聪的真实身份仍是未知数。请参考前言部分关于中本聪的更多讨论。



图7.3　比特币的市场价格（7天的平均值）

注：注意计算尺。

资料来源：bitcoincharts. com

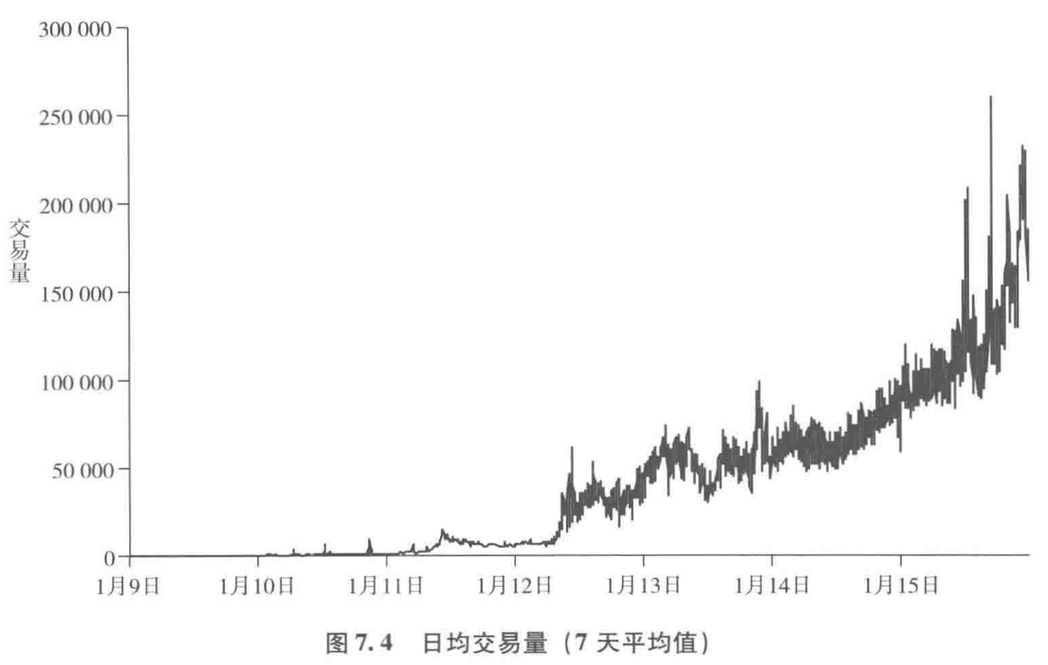


图 7.4 日均交易量（7 天平均值）

资料来源：bitcoincharts. com

7.5 政府对比特币的关注

本章接下来的内容我们将谈谈政府，政府与比特币的关系以及为监管比特币所做的努力。随着比特币成为一个足够庞大的现象，政府才开始注意到比特币，并开始担忧它可能造成的影响以及自己应该如何应对。本节和下节内容，我们将讨论为什么比特币会让各国政府产生担忧。在 7.7 节，我们将参照其他被监管的业务，讨论比特币可能会被监管的原因。7.8 节将对一个结合了常规消费者权益保护和比特币特性的监管提案进行案例分析。

资本管制

政府之所以注意到比特币这样的数字货币的原因之一在于，不可追踪的数字货币不受资本管制。资本管制是指一个国家颁布法律或法规，以控制流入或

流出该国的资本（货币或其他资产）。通过对银行和投资行为等设置限制性条件，国家可以监管资金流动。

在某些情况下，比特币可以轻而易举地绕过资本管制。人们可以在某国境内购买比特币，把这些比特币以电子方式转移到境外，然后将其兑现成资金或财产。这样，他们可以把资金或财产从境内转移到境外，或从境外转移到境内，而不受政府的管制。因为财产可以通过这样的电子方式轻而易举地进行跨境转移，而且无法真正被管控。政府如果想对比特币进行监管，就必须把比特币与当地法定货币银行系统隔离开来。这样，把大量当地货币兑换成比特币或反过来的做法就行不通了。一些试图进行资本管制的国家确实是这么做的，有些国家采取强硬措施，不允许企业用比特币换取当地货币，试图把比特币与法定货币银行系统隔离开来。

犯罪

不可追踪的数字货币让政府担忧的另一个原因是，它使某些犯罪更加容易，特别是涉及支付赎金的犯罪，例如绑架和勒索。如果可以从远程匿名支付，这些犯罪会更加容易。

例如，打击绑架的执法人员往往依靠受害者或其家人向绑匪转账的记录提供追查线索。远程匿名转账会加大执法人员追查转账流向的难度。还有一个例子，CryptoLocker 软件可以对受害人的资料恶意加密，并要求必须用比特币支付赎金才可以解密。这样，犯罪行为和支付行为都可以远程进行。同样，当人们可以很轻松地转账，参与的交易并不需要与某个特定的个人或组织绑定时，逃税对他们来说也更加方便了。最后，如果远程转移资金可以不通过监管机构，出售非法商品就会更加容易。

“丝绸之路”

丝绸之路公司（Silk Road）[①] 就是一个很好的例子，它自称“匿名市场”

① 丝绸之路公司是一个利用 Tor 的隐秘服务来运作的黑市购物网站。2013 年，美国 FBI 捣毁了丝绸之路公司，并且逮捕了该网站创始人。——编辑注

(anonymous marketplace)，被称为“销售非法药品的易贝网”。图7.5是“丝绸之路”的网站截图。毒品是它的主要销售商品，在网页右侧还可以看到为数不多的其他类别的在售商品。

卖家可以在“丝绸之路”宣传自己的商品，买家则可以购买它们。商品通常通过邮寄或货运的方式发到买家手中，支付方式为比特币。网站以Tor隐匿服务的方式运作，我们在第6章已经讨论过这个概念。从截图中我们可以看到，网站地址为http：//silkroadvb5piz3r. onion。这样，执法人员也无法追查到服务器的位置。

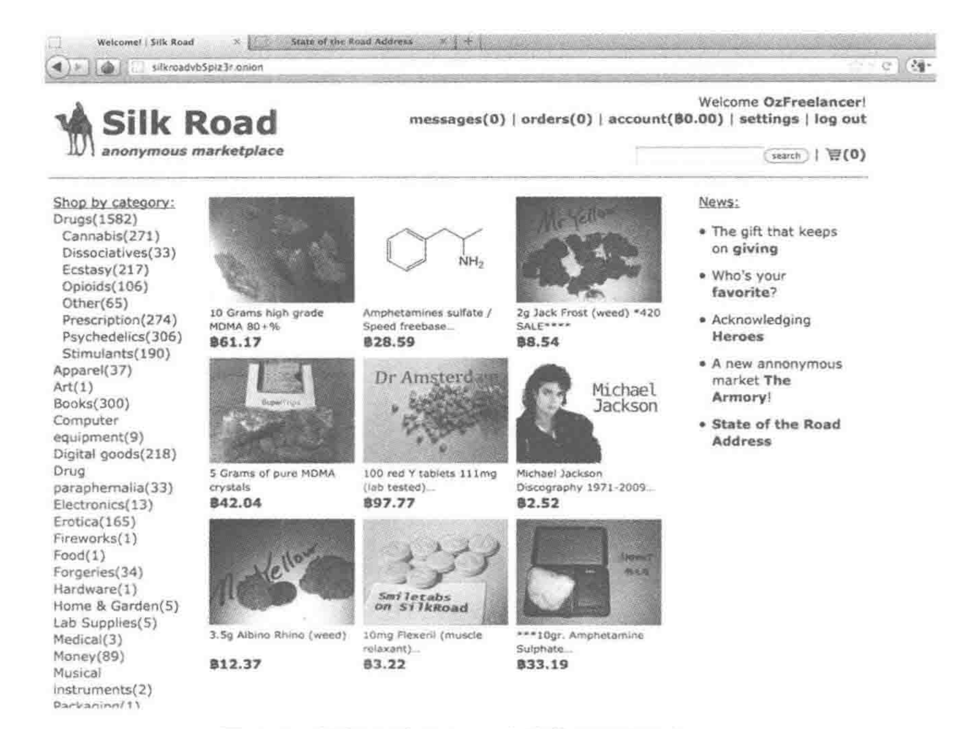


图7.5　丝绸之路公司网站的截图(2012年4月)

由于交易是通过比特币支付的，执法人员也很难追查资金流向，查出哪些人参与其中。商品发货时，“丝绸之路”网站暂为代管比特币。这个极具创新性的代管机制有助于防范买家和卖家被对方欺骗。买方确认收货之后，网站才会把比特币转给卖方。它还有类似于易贝的信誉评价系统（reputation system），买家和卖家在交易完成之后获得信誉评级。因为这个信誉评价系统，“丝绸之路”

鼓励买卖双方以符合市场规则的方式进行交易。因此，作为一个非法购物网站，“丝绸之路”是很有创新性的，它找到了一条以遵循市场规则进行远程非法交易的道路，而之前的非法市场很难做到这一点。

“丝绸之路”由一个自称“恐怖海盗罗伯茨”（Dread Pirate Roberts）的人运作，这显然是一个化名，极有可能来源于小说或电影《公主新娘》中主人公的名字。网站自2011年2月开始运行，2013年10月，它的运营者、后被认定就是“恐怖海盗罗伯茨”的罗斯·乌布利希（Ross Ulbricht）被逮捕之后，网站关闭。乌布利希曾试图通过使用多个化名账户、Tor及匿名重发（anonymous remailers）等方式掩盖自己的踪迹。尽管如此，美国政府还是将种种蛛丝马迹联系在一起，证据表明他与“丝绸之路”的活动、网站服务器以及作为网站运营者所控制的比特币之间存在密切联系。他被判犯下与运营网站相关的多项罪行。他的罪行还包括买凶杀人，幸运的是，他在这方面的能力实在太欠缺，并没有人被杀害。

在打击“丝绸之路”的过程中，联邦调查局收缴了约174 000比特币，当时市值超过3 000万美元。根据美国法律，政府有权没收任何犯罪所得收益。后来，政府拍卖了部分收缴的比特币。

“丝绸之路”的教训

从执法人员和“丝绸之路”的较量中，我们可以吸取一些教训。第一，把现实世界和虚拟世界完全分离开是很困难的。乌布利希认为自己可以既拥有现实生活，又可以用一个秘密身份来经营庞大的业务和技术设施。事实上，把这两个世界分开，让二者不存在任何一丝联系，是很难做到的。活跃地参与到一些需要与他人协调的活动中，同时想长期保持匿名，这会非常困难。一旦两种身份之间出现了一些联系，比如，如果你穿帮了，可能会用另外一个身份来为此打掩护，这种联系永远不会消失。这样，随着时间推移，一个人所使用的不同身份就会渐渐有了关联性。这正是发生在乌布利希身上的事情——他在早期用同一台计算机访问自己的个人账户和“恐怖海盗罗伯茨”账户。这些错误最终为调查人员提供了线索，发现他的现实身份。

另一个教训是执法人员可以对资金进行追踪。在乌布利希被捕之前，联邦调查局就已经知道“丝绸之路”运营者管理着的比特币地址，他们一直监控着这些地址。因此，尽管乌布利希的区块链拥有大量财富，他无法从这些财富中收益，因为只要他试图转移资产，就会留下可追踪的痕迹，很可能会导致他迅速被捕。因此，尽管乌布利希拥有174 000比特币，在现实生活中，他并没有过上奢侈的生活。他居住在旧金山一个一居室的公寓里，显然，他无法将自己积累的财富兑现。

简而言之，如果你有意经营一家秘密的犯罪公司——当然，我们并不推荐这条“事业”道路——做起来可能要比你想象中难得多。比特币和Tor这样的技术并不是刀枪不入的，执法部门仍然拥有很多重要的手段帮助侦破案件。尽管执法部门对比特币的崛起有过一定恐慌，但它们依然可以对资金进行追踪，而且也具备侦破犯罪案件的强大能力，让那些从事需要与他人协调的犯罪行为的人日子不那么好过。

同时，“丝绸之路”是关闭了，但执法部门并没有完全关闭通过比特币交易的经营非法药物的黑市。事实上，“丝绸之路”关闭之后，这类黑市如雨后春笋般涌现，其中一些较为突出的黑市包括绵羊集市（Sheep Marketplace）、“丝绸之路2.0”（Silk Road 2）、黑市重装上阵（Black Market Reloaded）、进化（Evolution）和阿格拉（Agora）。由于执法人员的行动或者内部人员的偷窃，其中大多数现在已经不复存在。虽然如此，研究发现，执法部门对这种黑市的打击并未使其放缓增长，其销售额反而上涨。网站经营者可能私吞买家被代管的资金，然后消失。为了规避这种安全风险，新出现的集市多采取多重签名代管（我们在第3章已讨论过），不再采用“丝绸之路”由经营者保管资金的方式。

7.6 反洗钱

这本节中，我们将讨论洗钱和反洗钱相关法律、政府，尤其是美国政府所制定的反洗钱（Anti-Money Laundering，简称AML）法律对比特币相关业务造成的影响。

反洗钱政策的目的在于防止资金的大量秘密外流或流入，防范资金在非法与合法企业或机构之间秘密流动。在7.5节我们讨论过，资本管制是为了防止资金的跨境流动。在某些情况下，政府并不介意资金跨境流动，但它们想知道资金的来源和去向。反洗钱政策的目的是给某些特定类型的犯罪行为增加难度，特别是有组织犯罪。有组织犯罪的罪犯经常会收到大笔资金，需要转移这些资金，但又不想对任何人解释这笔资金的来源，以及为什么想把这笔资金转移到境外。或者，罪犯通过非法活动赚到了一大笔钱，想通过合法企业使其合法化，这样，犯罪头目就可以购买各种奢侈品。反洗钱的目的就是让资金转移更加困难，而且，有人试图这么做时，会更容易被发现。

了解你的客户

“了解你的客户”（Know Your Customer，简称KYC）原则，是打击洗钱活动的根本对策之一。KYC原则的细节有些复杂，而且根据立场不同而有所变化，但其核心思想是，“了解你的客户”原则要求，对于一些有资金处理业务的特定类型的企业，需要做到以下三件事：

1. 识别并验证客户——获得客户的身份证明，确定客户的真实身份与他们声称的身份一致，并确保他们声称的身份与现实世界的真实身份对应。一个人推开门直接走进屋，说自己是住在美国某市某街道多少号的张三、李四，企业不能凭此口述确认他的身份，他必须提供可靠的身份证明文件。

2. 评估客户风险——确定客户从事地下行为的风险。通过评估客户行为以确定风险——他们与公司的合作关系是否长久、他们在社区的知名度以及其他多个因素等。通常而言，KYC原则要求公司对看起来风险更大的客户实施监控。

3. 监控异常举动——监控看似洗钱或犯罪的行为。如果一个客户看上去不甚可靠，或者无法对其身份进行充分验证，或者其行为没有符合法律的要求，那么KYC原则往往要求公司终止与这种客户的业务往来。

强制上报

美国的强制上报（mandatory reporting）对比特币的业务影响重大。众多行

业内的公司被要求必须将超过10 000美元的资金交易进行上报。它们需要提交一份《资金交易报告》，说明交易内容和交易对手的身份，这份报告还要求对交易对手进行身份验证。报告提交后，所有信息纳入政府数据库，以便将来分析是否存在洗钱行为的嫌疑。

美国政府还要求各公司关注那些可能通过人为操控以规避上报交易的客户，比如，由于法规只要求对金额超过10 000美元的交易进行汇报，一些人可能开展数笔金额为9 999美元的交易。如果发现客户操控交易，公司必须编写《可疑行为报告》，并汇报给政府。同样，这些信息也会被录入政府数据库，用于可能对客户开展的调查。

各个国家的要求各不相同。在此，我们并不是想提供什么法律建议。讨论这个话题只是为了说明反洗钱法规提出的一些要求。不管是美国政府还是其他政府，都非常重视反洗钱法规，一旦有人违反，可能会被判以重刑。有一些法律，违反之后，政府可能会投诉你，然后你再设法去解决它，反洗钱法并不是这样的。

政府已经关闭了许多比特币企业，一些是暂时的，一些是永久的。还逮捕了一些从业人员，也有一些人因为违反反洗钱法而锒铛入狱。不管是使用比特币还是法定货币，国家都会严厉打击洗钱行为。自从注意到比特币市场已经足够大，可能会带来洗钱风险，政府部门就强化了对比特币相关企业可能洗钱的监控。如果你有兴趣创建一家需要处理大额资金的企业，那就需要跟理解这些法规的律师好好谈谈。

7.7 监管

现在让我们来直接谈谈监管问题。监管通常背负恶名，亲比特币派尤其不喜欢这个词。这些人认为，监管是一种官僚主义，它们根本就不了解我的业务，不懂我做的事情，只会横插一脚，把事情搞砸。监管是一种负担，它不仅愚蠢，而且毫无意义。这种观点很普遍，而且容易得到认同，它也许不无道理，但此

处我们不再重复了。

相反，在本节中，我们将探讨为什么监管有时是可取的，因为很多人并不理解这种观点。澄清一点，我们虽然在本节要花大量篇幅来解释为什么监管有时是一件好事，这并不代表我们支持对比特币进行全面监管。但是我们想让认为监管生而可恶的社区听到一点不同的声音。

赞成监管的基本原因是：当市场失灵并带来大家一致认可的恶果时，监管可以介入，并解决这种失灵。由于市场并不总是给出最优的结果，所以我们说监管有时是有益的。

我们可以用经济学术语更为准确地表达这一观点。我们担心市场会失灵，“失灵”的意思，并不是说坏事发生，或者有人觉得他们被敲诈或被不公平对待。我们指的是，对市场参与者进行不同的商品分配，这会让每个人更好，或者至少不会更差。这种不同的分配被称为帕累托改进（Pareto improvement）。

柠檬市场

柠檬市场（lemons market，也称为次品市场）是可能导致市场失灵的一个经典案例。柠檬市场的名称起源于汽车销售行业，但这个概念并不局限于这个行业。假设所有汽车要么优质要么劣质，没有中间地带。制造一辆优质汽车的成本比一辆劣质汽车的成本高不了多少，但对于购买汽车的人来说好处则多很多。

如果市场运行良好（即经济学家所说的“有效运行”），汽车销售商会向顾客提供更多优质汽车。这是因为，虽然优质汽车价格较高，但大多数消费者更喜欢优质汽车，愿意花更多的钱购买。所以，在假设条件下，市场会提供这样令人愉悦的结果。

但是假设顾客分辨不出哪些汽车是优质的，哪些是劣质的。一辆劣质汽车（一个“柠檬”，俗称次品）从外观上看上去似乎很不错，但顾客不知道它会不会明天就熄火，还是可以开很久。销售商很可能知道它是一个次品，但是顾客是分辨不出来的。

接下来，我们考虑这种柠檬市场是如何驱动人们的消费行为的。作为一名

消费者，由于在购买汽车前根本看不出区别，即便销售商告诉顾客这辆车非常好，顾客只需要多掏100美元就可以买下，顾客未必会相信他，也就不愿意掏多余的钱买一辆优质汽车。

由此带来的结果是，生产商并不会因为多卖出一辆优质汽车而多赚一笔钱。事实上，每卖一辆优质汽车，它们反倒赔钱，因为优质汽车的生产成本更高，它们并没有赚到差价。最终，生产商只生产劣质汽车，而顾客对自己所购买到的商品非常不满，市场就卡在这个平衡上。

与正常运行的市场相比，这个结果对每个人来说都更糟糕。它对消费者来说更糟，因为他们不得不凑合使用劣质汽车。在一个运行更加有效的市场中，他们可能只需要多花一点钱，就可以购买到质量好得多的汽车。对于生产商来说也很糟，因为市场中出售的汽车都是次品，消费者可能就不会买那么多汽车，因此，销售汽车获得的利润要比在一个健康市场上的利润少得多。

这个现象就是市场失灵。柠檬市场并不专指汽车行业。任何待售的存在信息不对称（asymmetric information，即买方或卖方中的一方比另一方对商品品质的了解要多）的商品或小部件都会遭遇市场失灵。经济学文献可以提供汽车行业以外的更多案例。

修复柠檬市场

通过一些市场手段，可以修复柠檬市场。一种手段是通过卖方的信誉。如果卖方总是实话实说，告诉买方哪些是优质品哪些是劣质品，久而久之，这个卖方就会被大家认为是诚实的。一旦有了这种信誉，它们就能够以更高的价格出售优质汽车，因为消费者会相信它们说的话，市场也就可以更有效地运行。这一手段有时奏效，有时无效，取决于对市场所做的假设是否准确。当然，它即便有效，也比不上消费者能够分辨商品质量好坏的市场。而且，生产者需要一段时间才能建立自己的信誉。这也就要求它们必须要在一段时间内以低价销售优质的商品，直到消费者意识到它们的诚实可靠。这加大了诚信卖家进入市场的难度。

还有另外一个潜在问题，即便是一个一直有诚信的卖家，当它决定退出市

场（比如因为销售额下降）时，就不再有动力继续对买方说实话。它可能会趁机大肆欺骗买方，然后关门大吉。因此，在卖家一开始进入市场或决定退出市场时，通过卖家信誉来修复柠檬市场的方式并不一定奏效。

对于消费者不会从同一个买家重复购买商品的行业，或者商品是新兴事物，卖方还没有足够的时间建立起信誉的行业，通过信誉修复柠檬市场的手段也很可能不会奏效。像比特币这样的高科技市场就面临这样的问题。

另一种修复柠檬市场的手段是担保。卖家向买家提供担保，也就是，如果商品最终被证明是劣质的，卖家会换货或退款。这种方法是比较有效的，但也存在一个问题：担保也是另一种可能存在质量高低的商品！如果是一个劣质担保，当购买的商品出现问题后，卖家可能根本就不兑现之前的承诺，或者在买家要求兑现担保承诺的过程中故意设置各种难题。

通过监管手段修复

如果确实存在一个柠檬市场，而且上述所有的市场手段均未奏效，那么监管可能会帮上忙。具体来说，监管可以通过以下三种方法修复柠檬市场。

第一种方法是，监管可以要求信息公开。比如，可以要求所有的汽车都贴上标签，标明它是优质汽车还是劣质汽车，并对造假的企业施以处罚。这可以让消费者了解他们之前并不了解的信息。第二种监管方法是出具质量标准，只有通过质量标准检测的汽车才可以出售，否则不得出售。有了这样一个标准，只有优质汽车才能够通过质量检测。如果监管奏效，可能会导致市场上只有单一质量的汽车，但至少都是优质汽车。第三种方法是，监管可以要求所有的销售方出具担保，并强制执行这些担保，这样，销售方就必须对其做出的承诺负责。

所有这些监管手段都可能失效，它们可能达不到预想效果，可能会写得不好、误用，或者对卖家造成负担。但是这种监管为柠檬市场导致的市场失灵问题提供了一个可能的解决方案。例如，一些支持对比特币交易市场进行监管的人，有时就认为它是个柠檬市场。

串谋和反垄断法

市场不以最优方式运作的另一个例子就是价格垄断。价格垄断是指不同卖家相互串通，一致上调或下调价格的做法。另一个与之相关的情况是，本应该是竞争关系的公司决定不再相互竞争。例如，某市有两家面包店，它们商量好，一家只卖松饼而另一家只卖面包圈，这样，比它们两家同时既卖松饼又卖面包圈的竞争要小得多。竞争减少，商品价格自然上涨，商家对市场的运行造成阻碍。

总之，一个正常运转的市场主要通过竞争来保护消费者权益。卖家必须以最优惠的价格向顾客提供最好的产品来进行竞争，否则，它们就没有业务。价格垄断或串谋行为规避了竞争。如果人们采取措施规避竞争，这是另一种形式的市场失灵。

在大多数司法管辖区，商量好涨价或者不竞争的行为是违法的。这是反垄断法或竞争法的一部分内容。这一法律的目的是限制蓄意阻碍或损害竞争的行为。一般来说，它更侧重于限制诸如通过并购减少竞争的行为，而不是为消费者提供物美价廉的商品。反垄断法很复杂，我们只做了一些简单介绍，但它为市场失灵后，法律如何介入并纠正市场失灵提供了一个案例。

7.8 纽约州比特币牌照

截至目前，我们已经对监管内容做出了概述：监管的不同形式，为什么在某些情况下监管是有利于经济运行的。接下来，我们来看看纽约州比特币牌照（Bitlicense），这是某个州对比特币监管所做出的一项具体努力。此处描述的细节对于我们的讨论影响不大，因为我们的目标并不是让你了解一个法律条款。相反，我们希望帮助你了解监管部门都在做哪些事情以及它们是如何考虑这些问题的。

《纽约州比特币牌照提案》最初于2014年7月提交，后续根据比特币社区、

行业、公众以及其他利益相关者的反馈进行了修订，最后由州立监管金融行业的纽约州金融服务管理局（New York Department of Financial Services，简称NYDFS）于2015年8月颁布施行。当然，纽约州是世界最大的金融中心，州金融服务管理局也习惯与大型金融机构打交道。

涉及范围

《纽约州比特币牌照提案》是一套关于虚拟货币（virtual currency）的规范、法规和规章制度。它要求，如果你想要从事以下任何事情，那么必须从纽约州金融服务管理局获得所谓的比特币牌照：

> 虚拟货币业务行为是指涉及纽约州或纽约州居民的以下类型的行为：
>
> 1. 接收虚拟货币用于传输或传输虚拟货币，交易用途为非金融性且不涉及超过虚拟货币面额的转移除外。
> 2. 存储、持有或为他人代管虚拟货币。
> 3. 为客户提供购买或销售虚拟货币的服务。
> 4. 为客户提供兑换服务。
> 5. 控制、管理或发行虚拟货币。
>
> 内部员工或其本身对软件的开发与传播，并不属于虚拟货币业务行为。
>
> （摘自纽约金融服务管理局的《纽约州比特币牌照提案》的原文。）

以上条款是指“涉及纽约州或纽约州居民的行为”，反映了纽约州金融服务管理局的监管权力。然而，这种法规不仅对纽约州，对其他州也造成影响，原因有二：第一，在面临要么遵守州的法律，要么放弃在该州的业务时，像对纽约或者加利福尼亚这样人口众多的州，大多数公司都会选择遵守州法。第二，在对一些经济领域的监管上，一些州通常被视为领导者——纽约州在金融领域、加利福尼亚在科技领域。这也就意味着，美国其他州会沿着领导者设定的方向走下去。

请注意第一条提到，“交易用途为非金融性的除外”，这是在第二版修订中增加的，这一点加得很好，它剔除了那些仅把比特币当作平台的应用程序，我

们将在第9章探讨。第二条涵盖了钱包服务。至于第三条，你可以为自己购买和出售比特币，但是把它作为业务向顾客提供服务则需要获取比特币牌照。第四条足够清晰明白。最后一条可能更适用于另类币，与比特币相比，许多另类币的中心化更强。我们将在第10章中讨论另类币。

在条款的最后，把软件开发作为例外特别声明，也是非常重要的。在最初版本中并不包括这句申明，引发了比特币社区的强烈抗议。纽约州金融服务管理局局长本杰明·劳斯基（Benjamin Lawsky）随后做出解释，申明本条款的目的并不是为了监管开发人员、矿工或使用比特币的个人。最终版本包含了上述两个明确的例外情况。

要求

相关实体都必须申请一个牌照。你可以在提案中找到关于如何申请牌照的详细规定（参见本章最后的“延伸阅读”），但是简而言之，你需要提供一些充分的材料，证明你对企业的所有权、经济状况、保险和商业计划，以便让纽约州金融服务管理局了解你是谁、你是否有足够诚信、你的经济来源以及你打算用此做什么。此外，你还需要支付一笔申请费。

获得牌照后，你还需要向纽约州金融服务管理局提供所有权、经济状况、保险等信息。你还必须定期提供财务报表，以便让它们了解你的经营状况。你还需要维护一笔财务储备金，纽约州金融服务管理局会根据你的业务具体情况，确定具体金额。

提案包含如何监管客户资产等内容，也有非常详细的条款。提案也包括反洗钱条款，其内容可能与现有法律一致，也可能比现有法律规定的内容更多。提案包括关于安全计划和渗透测试等方面的条款；还有一些关于灾难恢复预案的条款，规定了必须制订灾难恢复计划以应对一切可能发生的糟糕情况；包括了历史记录保存的相关条款，申请者必须要保存记录，并允许纽约州金融服务管理局在某些情况下对其进行访问；申请者还必须制定合规的章程，在组织内部任命专门的合规员，并赋予必要的权力，确保业务的合规性。此外，申请者还必须向客户披露风险，让他们了解与申请者进行业务往来可能存在的风险。

正如你所看到的，提案要求的名目繁多，与成立共同基金或股票上市所要满足的要求极为类似。因此，比特币牌照是比特币历史上关键的一步。可能还有其他部门也会开始介入比特币的监管，然后比特币业务就会越来越接近传统的受监管的金融业务。

这可能会跟密码朋克和密码自由主义者们对比特币的期望背道而驰。但这可能具有一定的必然性，因为随着比特币价值的增加，比特币业务将会越来越大，政府会对它们产生兴趣，监管也就随之而来。比特币业务会对现实世界的人们及实体经济产生影响。如果比特币发展到了这种程度，也就意味着它已经发展到了需要被监管的程度。它表明比特币最初簇拥者的理念开始淡出，但另一方面，它也表明比特币生态系统在不断壮大，并且正在与受到更严格监管的实体经济不断融合。不管你对此持何种态度，对比特币的监管正在发生，如果你有兴趣创立一家比特币公司，你需要关注这一趋势。

这种监管比特币的努力会成功吗？可以有不同的方式来看待它，但是有一种方式，可以从提升比特币业务质量的角度，来评估像比特币牌照这种监管措施的有效性：如果企业在向非纽约州的客户推广业务时申明，它们拥有比特币牌照，因此它们是可以信赖的。假如企业的申明会让客户信服并由此开展业务往来，那么监管措施正如它的支持者设想的那样，发挥了作用。这个场景是否会发生，以及监管措施究竟会产生什么样的影响，让我们拭目以待。

延伸阅读

关于“丝绸之路”及其后继者的运作模式的两篇论文：

Christin, Nicolas. “Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace. In *Proceedings of the 22nd International Conference on the World Wide Web*, New York: ACM, 2013.

Soska, Kyle, and Nicolas Christin, “Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem.” In *Proceedings of the 24th USENIX Security Symposium*, Berkeley, CA: USENIX, 2015.

以下是比特币监管问题指南：

Brito, Jerry, and Andrea Castillo. *Bitcoin: A Primer for Policymakers.* Fairfax, VA: Mercatus Center at George Mason University, 2013.

一本讲述比特币社区及其主要特征的非技术性著作：

Popper, Nathaniel. *Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money.* New York: Harper, 2015.

一篇阐述数字货币的早期作品，探讨了未来世界如何保护数字隐私问题：

Chaum, David. "Security without Identification: Transaction Systems to Make Big Brother Obsolete." *Communications of the ACM*, 28 (70), 1985.

一项对信息安全经济学的调查，其中讨论了市场失灵的一些原因：

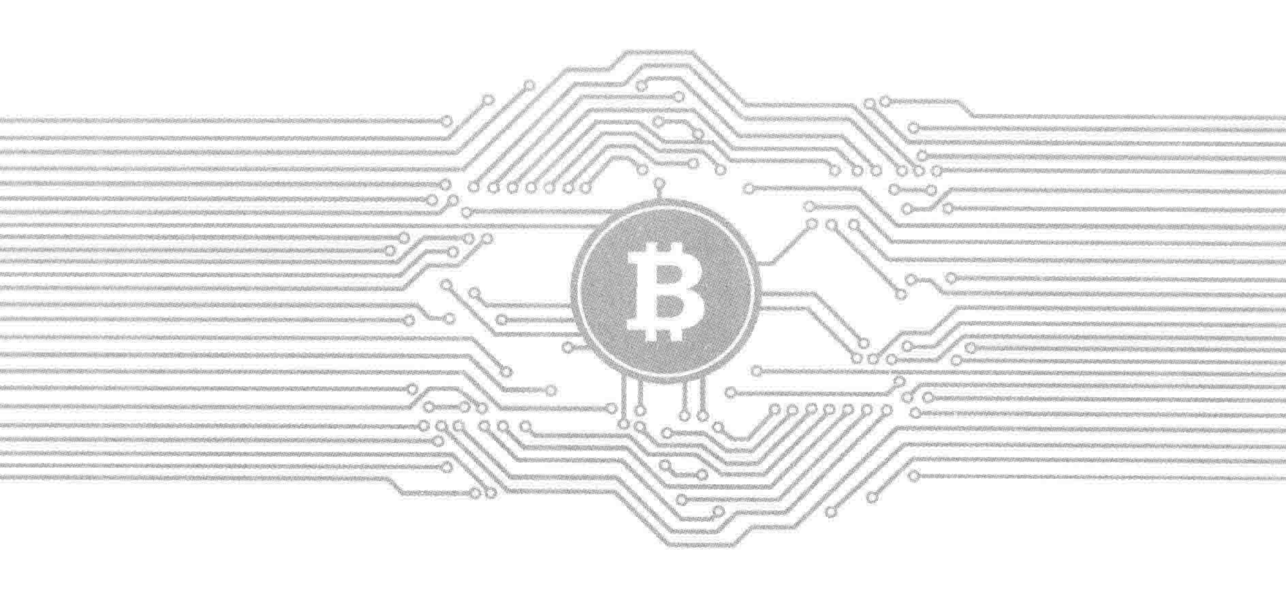
Anderson, Ross, and Tyler Moore. "The Economics of Information Security." *Science* 314 (5799), 2006.

讨论比特币的经济问题和监管方案：

Böhme, Rainer, Nicolas Christin, Benjamin Edelman, and Tyler Moore. "Bitcoin: Economics, Technology, and Governance." *Journal of Economic Perspectives* 29 (2), 2015.

《纽约州比特币牌照提案》文本：

New York State Department of Financial Services "Regulations of the Superintendent of Financial Services. Part 200: Virtual Currencies." 2015. 下载地址：http://www.dfs.ny.gov/legal/regulations/adoptions/dfsp200t.pdf.



第8章

其他挖矿算法

BITCOIN
AND
CRYPTOCURRENCY
TECHNOLOGIES
A Comprehensive Introduction

由于挖矿算法的复杂性使得任何个人或团体都难以操控共识成形的过程，挖矿算法被认为是比特币系统的核心。因为比特币矿工通过解谜来获得奖励，所以我们可以期望他们会花大量的时间与精力去寻找捷径而更加快速有效地解谜，以增加他们的收益。另一方面，如果有些工作对网络有利但并不能让矿工更快速地解谜，他们可能会忽视那些工作来最小化他们的成本。所以解谜的设计对引导和指引矿工起着至关重要的指导作用。

在本章中，我们会讨论一些其他的挖矿解谜（mining puzzle）设计，假设我们可以改善比特币的解谜甚至重新设计一套新的解谜过程。一个经典的设计挑战是让解谜过程能够限制 ASIC 挖矿，这样一来可以平衡计算机设备性能上的差距（拥有一般电脑的矿工与拥有优化过的 ASIC 矿工之间的设备差距）。还有什么其他设计是需要我们考量的？有哪些行为需要我们鼓励，而哪些需要阻止？我们会讨论一些有着不同特征的案例，从减少能源消耗（这对社会发展有着积极意义），到约束挖矿工具的形成。有一些已经被另类币所采用，另外一些还处于理论研究阶段，可能将来会被用到。

8.1 算法的基本要求

我们首先来看一下一些挖矿算法的主要安全要求。如果算法本身不能满足比特币安全性上的基本要求的话，我们也没有必要引入一些新奇的特点。

已经有许多可能的要求，有些我们在前面的第2章和第5章中已经讨论过。挖矿解谜的结果需要被及时验证，因为每个在网络上的节点都在验证每个解谜的结果，即使是那些没有直接参与挖矿的节点，包括SPV（简单支付验证）的客户端。我们还需要解谜的难度具有可调整的特征，解谜难度可以随着新加入用户而增大的哈希算力得到调整。这样一来，解谜过程就可以具备足够的难度使得对区块链的攻击变得代价高昂，同时又能保证解谜本身可以在一个稳定的频率上实现（比特币系统中大约每10分钟完成一个解谜过程）。

到底什么是比特币的挖矿解谜？

到现在为止我们一直在用"比特币解谜"这个名称，更加精确的说法是，我们称它为一个"不完全哈希函数原像解谜"（partial hash-preimage puzzle），因为这个运算的目的，是找到一个不完全的特定哈希函数输出值的原像——也就是一个低于某一特定目标区值的结果。除此之外，一些罕见的特征也可以用来作为比特币的挖矿解谜运算，比如找到一个区块，它的哈希函数值至少有k个点位是零，但是通常直接比较既定目标是最简单的方法。

比特币用的基于SHA－256挖矿解谜哈希函数，很显然已经满足了这两个要求。它可以通过任意调节一个参数（目标）来灵活增加难度。检查这个谜底很容易，只需要一个SHA－256计算和一个与目标的比较即可，不管找到这个谜底的过程有多么困难。

另外一个核心的要求更加微妙：在任意单位时间找到一个谜底的成功率，大致上要与所贡献的哈希算力成比例。这就意味着，大矿工虽然拥有非常强大的挖矿机，他也只是有着一定比例的优势来成为下一个找到谜底的矿工。即使是小矿工，也会有一定的机会能够成功并且获取奖励。

为了说明这一点，我们先来设想一个没有满足这个要求的不合格解谜过程。想象一下某一个挖矿解谜要经过精确的n个步骤找到一个谜底。例如，不同于我们当前要求的"找到一个SHA－256结果低于某一个固定目标的区块"的做法，如果要求计算n个连续的SHA－256函数值，这种做法检查结果会变得没有效率，但是这个问题目前无关紧要，更大的问题在于，因为这个解谜过程需要

精确的n个步骤来完成，所以网络上解谜更快的矿工将会永远是获得下一个奖励的赢家。很快这个情况就变得路人皆知，最快的矿工会完成所有解谜，而其他矿工完全没有动力继续参与下去。

再次声明，一个好的解谜方案，是给每个矿工一个按比例性的成功概率来赢得下一个谜底，这个概率是与他们所贡献的哈希算力成比例。就好比往一个不同大小色块组成的目标板上随机地掷飞镖，每个不同大小色块就类似于不同矿工所具有的挖矿运算能力。如果你考虑到这一点，这就意味着你猜中谜底的概率并不取决于你已经做了多少工作去解谜（因为大矿工们总是会做更多的工作量）。所以一个好的解谜是“无关过程的”（progress free）[①]。

从数学角度来看，一个好的挖矿解谜一定是一个“无记忆进程的”（memoryless process）——而任何其他的方法都将由于过去的挖掘工作，不可避免地在一定程度上奖励挖矿工人。因此，任何可行的解谜从根本上都是一个不断试错的过程（trial-and-error）。这种解谜所需要的时间，必然服从一个指数分布[②]，我们曾在第2章讨论过。

可以调整的难度、快速验证和无关过程属性，是比特币挖矿解谜的三大核心特征。基于SHA-256算法的“不完全哈希函数原像解谜”显然满足了这三大要求。有些人可能会说其他一些特征也很重要，我们在后面讨论其他潜在功能的时候会提及。

8.2 反ASIC解谜算法

首先我们从讨论设计一个可以反ASIC解谜（ASIC-resistant puzzles）的挑战开始，这个挑战也是最被广泛讨论和追求的可替代目前比特币挖矿解谜的一种。我们在第5章中讨论过，比特币挖矿最初是用普通电脑，然后再升级到GPU和

① 意思是来得早，不如来得巧，但这个巧后面的学问就大了。——译者注

② 旅客进入机场的时间间隔也是一个指数分布，后面进来一个人的时间间隔与前面进来人的时间间隔无关。——译者注

定制化的 FPGA 设备，到现在基本上由非常强大的优化过的 ASIC 芯片所垄断。现在的 ASIC 的挖矿运算能力比一般电脑甚至早期的 ASIC 都要高太多。一般的电脑即使硬件本身是免费的，也会因为电费价格等因素而变得不可行。

这个转变意味着，在比特币生态系统里的大部分个体（例如使用比特币交易的客户和商家）已经无法参与到挖矿过程中了。有些人认为这是一个危险的势头，一小部分职业矿工控制了整个挖矿的过程。在中本聪最初有关比特币的论文里，用到过"一个 CPU 一票"的说法，这个说法时不时被有些人用来说明比特币应该是一个被全部用户所拥有的民主系统。

其他人觉得 ASIC 的崛起是不可避免的，而且这也不会伤害到比特币，这种希望实现反 ASIC 的愿望也只是有些人希望回到"过去的好时光"。对于反 ASIC 是否可取，我们保持中立的态度，因为只有这样，我们才可以深入讨论一些技术上的挑战和提议的方案，来实现反 ASIC 的目标。

反 ASIC 到底是什么意思

大致上说，我们想抑制为了挖矿而特别定制的设备的优势。这也可以理解为，设计一个解谜程序，让现有的普通电脑成为最廉价和最有效率的解谜运算设备。但这在现实中不可能，毕竟所有的通用电脑的中央处理器里已经针对一些特殊目的进行了优化。并不是所有的电脑都有相同的优化配置，并且它们随着时代而改变。比如，过去的 10 年中，英特尔（Intel）和 AMD 在芯片里加入特殊指令（通常叫作"增加硬件支持"）来更加有效地计算高级加密标准（Advanced Encryption Standard，简称 AES）的区块密码。所以有些电脑在挖矿这个事情上总是会比其他电脑更加低效。另外，很难想象设计一个挖矿解密程序，而这些程序是依赖普通个人电脑诸如音响或显示器这些特性的，所以去除了那些通用特性的具有特殊目的的设备，很可能会更有效率，并且成本更低。

更加实际地说，我们有一个适中的目标：设计一个解谜程序，尽可能地减少最有效率的定制运算设备与通用电脑之间的效率差距。ASIC 还是会不可避免地成为更加有效的挖矿机，但我们至少可以将其运算效能限制在一定范围内，从而让个人用户使用他们已有的通用电脑来挖矿仍具备一定的经济效应。

刚性内存解谜

大多数被设计成反 ASIC 的解谜程序中，最普遍被应用的叫作刚性内存解谜（memory-hard puzzles）——解谜需要大量的内存来计算，而不是靠大量的 CPU 时间。一个类似但又不一样的概念是内存限制解谜（memory-bound puzzles），花在读取内存上的时间，占据了这种程序大部分的计算时间。一个解谜可以是刚性内存类而不是内存限制类，或是内存限制类而不是刚性内存类，或是二者兼而有之。一个微妙但重要的区别在于，虽然 CPU 的速度是计算时间的瓶颈，但平行运算大量解谜的成本，还是被内存的成本所左右，或者反之亦然。通常对于运算类的解谜程序，我们要做到刚性内存和内存限制，就需要保证在运算过程中大量的内存被要求使用，使之成为一个限制性因素。

为什么刚性内存解谜或内存限制解谜可以反 ASIC？因为用来计算哈希函数的逻辑运算只占了 CPU 里的一小部分，意思是在比特币的解谜计算里，ASIC 不需要执行一些不必要的功能，而只需要执行计算哈希函数的相关功能，所以占了很大优势。另外一个相关因素是，不同的内存性能上的差异（和单位性能的成本）比不同处理器之间运算速度上的差异要小很多，所以，如果我们设计了一个刚性内存类的解谜，计算时需要相对简单的算力但需要大量的内存，这就意味着，解密成本的上升速率将会像内存成本提升速率那样，在一个相对低一些的水平。①

SHA－256 已经被认定为不是刚性内存解谜算法。它只需要一个小小的 256 位模块，可以很容易地被放进 CPU 的注册机里。但设计一个刚性内存类的工作量证明解谜不是一件太难的事。

Scrypt

现在最受欢迎的刚性内存解谜叫作 Scrypt②，被第二大加密数字货币莱特币

---

① 也就是花费更多去提高内存的效能，并不能以相同比例去提高解谜的效能。——译者注

② Scrypt 是由著名的 FreeBSD 黑客 Colin Percival 为他的备份服务 Tarsnap 开发的。Scrypt 不仅计算所需时间长，而且占用内存也多，使得并行计算多个摘要异常困难，因此利用 rainbow table 进行暴力攻击更加困难。Scrypt 没有在生产环境中大规模应用，并且缺乏仔细的审察和广泛的函数库支持。——译者注

以及其他加密数字货币所用。

Scrypt 是一个刚性内存的哈希函数，最早是为了加密密码而不容易被暴力破解（比如，反复试错破解），所以挖矿解谜与比特币用的“不完全哈希函数原像解谜”是一样的，只不过用 Scrypt 取代了 SHA－256。

Scrypt 在比特币被发明出来之前就已经存在，而且它是用来加密个人密码，这一点让我们对它的安全性有一定的信心。密码的哈希函数化其实与反 ASIC 有着相似的目的。出于安全性考虑，我们期望，一个有着定制化设备的攻击者不能够比使用一般电脑或者服务器的用户更快地计算密码的函数值。

Scrypt 基本上有两个步骤：第一个步骤是在用随机数据填充随机存取存储器（Random Acess Memory，简称 RAM）里面的缓存空间；第二步是从这块内存区域里虚拟随机地读取（或者更新）数据，同时要求整个缓存都存储在 RAM 里面。

```
1  def scrypt(N, seed):
2      V = [0] * N  //初始化N长度的缓存区域

     //往这个区域里充满虚拟随机数
3    V[0] = seed
4    for i = 1 to N:
5       V[i] = SHA-256(V[i-1])

     //然后从这个区域里虚拟随意地读取
6    X = SHA-256(V[N-1])
7    for i = 1 to N:
8       j = X % N  //根据X，选择一个随机的索引
9       X = SHA-256(X ^ V[j])  //根据X的索引来更新这个X

10   return X
```

图 8.1　Scrypt 虚拟代码

图 8.1 展示了一段 Scrypt 的伪代码（pseudocode）来体现核心的计算原则，但我们也省略了一些细节——在实际中，Scrypt 使用更大一点的数据块，然后用来充满缓存区的算法略微复杂一点。

为了理解为什么 Scrypt 是刚性内存类的，我们先想象一下如果我们要计算

同样的值，但不用缓存区 V（参见图8.1）。这当然也是可行的——但在第9行代码里，我们需要重新动态地计算值 V [j]，这需要进行 j 次的 SHA-256 的迭代运算。因为 j 的值在每次迭代循环里会从 0 和 N-1 中虚拟随机地选择，因此这平均需要 N/2 次 SHA-256 计算。这意味着计算整个函数需要 $N \times N/2 = N^2/2$ 个 SHA-256 计算，但是如果使用一个缓存的话，只需要进行 2N 次运算。因此，缓存的使用将 Scrypt 的时间复杂度从 O（N^2）转换成 O（n）。这样一来，只要简单地选一个足够大的 N 值使得 O（N^2）的计算变得足够慢，以此确保使用内存是更快的选择。

在时间与内存之间的权衡

如果没有一个较大的内存缓存，计算 Scrypt 会变得很慢，但是用较少的内存来增加相对较少的计算还是可能的。假设我们使用一个大小约 N/2 的缓存（而不是 N 的大小），现在，我们只在 j 是偶数的情况储存 V [j] 的值，丢掉那些 j 是奇数的值。而在第二次循环里，一半的情况下 j 为奇数的值将会被选到，但这种情况还是很容易被计算的。我们只需要简单地计算 SHA-256（V [j-1]），因为 V [j-1] 在我们的缓存里。[①] 在一半的时间内会产生这种情况，所以它增加了 N/2 个额外的 SHA-256 计算。

因此，对内存要求量的减半只会增加 1/4 的 SHA-256 计算量（从 2N 到 5N/2）。总体来说，我们可以储存缓存区域 V 里的每个 k 排数据，即使用 N/k 的内存和计算（k+3）N/2 次的 SHA-256 迭代计算。在这个限制下，如果我们设定 k=N，我们就回到先前运算时间为 O（N^2）的计算。这些数字不一定非常精确地适用于 Scrypt 算法本身，但是渐近预测的方式确实是适用的。

除此之外，还有其他的设计可以弱化用时间来换取内存的能力。举例来说，如果一个缓存持续地在第二次循环中被更新，它可以让时间与内存之间的互换不是那么有效，因为这些更新必须被储存在内存中。

---

① j 是奇数时，减 1 为偶数，我们存的是有偶数的值的。——译者注

校验成本

Scrypt 的另一个局限性是，它需要用与计算所用的同样大小的内存来做校验。为了让内存刚性有意义，N 需要变得比较大。这意味着一个 Scrypt 的计算要比一个 SHA－256 的迭代计算（在比特币里只需要一个 SHA－256 计算就可以校验）昂贵许多倍。

这会产生负面的结果，因为在网络里的每个用户必须重复这个计算来检查每一个新发现的区块是否有效。这会减缓新区块传播和被认可的速度，从而增加了分叉攻击[①]的风险。它还要求每个客户端（即使是轻量级的 SPV 客户端）拥有足够的内存来有效地进行函数计算。这样一来，实际上在加密数字货币中能够被 Scrypt 用到的内存 N 是有限的。

一直到最近我们都不明确，是否有可能设计一个挖矿解谜程序在计算上是刚性内存类的，又可以很快地（不需要大量内存）进行校验。这个特性对密码进行哈希运算没有多大作用，在用于加密数字货币之前，这是 Scrypt 算法的主要用途。

在 2014 年，一个叫作杜鹃鸟周期的新解谜算法被约翰·特龙普（John Tromp）所提出（起这个名字是因为这个算法的特性与杜鹃鸟的特性类似，杜占雀巢）。杜鹃鸟周期算法，是从杜鹃鸟哈希表所衍生的一张图中寻找周期的难度而设定的，杜鹃鸟哈希表这种数据结构在 2001 年才被首次提出。除了建立起一个很大的哈希表之外，没有其他已知的方法来计算这个周期，结果却可以通过发现一个周期（相对小的）来简单地验证。

这个算法可能会让刚性内存或是内存限制类的证明工作在比特币共识里变得更加实用。可惜的是，这个函数无法在数学上证明，如果它不用内存的话就不能被有效地计算。通常，一个新的密码学算法看起来都是安全的，但是社区会对它持有保留意见，直到它存在了多年而没有被破解过。因为这个缘故，并且因为它也是最近才被发明的，当前杜鹃鸟周期算法还没有被任何加密数字货

① 有关分叉攻击的内容，可以参见本书第 5 章。——译者注

币所采用。

实际应用中的 Scrypt

Scrypt 被许多种加密数字货币所使用，包括莱特币在内的几种热门币，结果好坏参半。针对莱特币 Scrypt 算法参数的 ASIC 已经存在（然后被其他几种另类币所复制）。令人惊讶的是，相较于大众电脑，这些 ASIC 在算力上的提高比起 SHA-256 相对普通电脑的提高，至少旗鼓相当甚至要更大！所以，Scrypt 最终还是无法反 ASIC，至少在莱特币上是如此。莱特币的设计者起初宣称反 ASIC 是莱特币的一大优势。但现在他们已经收回了这个说法。

这可能是莱特币所用的数值 N（内存使用参数）比较低所造成的，它只要求 128KB 就可以进行计算（如果使用时间内存互换的模式，可能所需要的内存更低，这也被普遍用于 GPU 以获得更快的缓存）。低数值 N 使设计一个不需要复杂的内存存储总线的轻量级挖矿 ASIC 变得很容易，通常这种复杂的总线是读取十亿字节（Gigabytes）级别的随机存取存储器所需要的，而这些通用电脑都具备。莱特币的开发者没有选择一个比较高的内存参数（这会使 ASIC 更加难以设计），因为他们认为高内存参数所导致的高成本的校验过程是不太现实的。

其他抵抗 ASIC 的方法

请回忆一下，我们的初衷是想让可以大幅度提升计算性能的 ASIC 的开发变得困难。刚性内存解谜只是其中一个方法，还有其他方法。

遗憾的是，其他的方法都不是很科学，并且没有作为刚性内存函数而被设计过或者攻击过。最有名的一个叫作 X11，其实就是把 11 个不同的哈希函数结合在一起，被一个叫作“黑暗币”（Dark Coin）的另类币所用（后面这个另类币改名叫 DASH），在 DASH 之后也被其他一些另类币所使用。X11 的目的是使设计一个有效的 ASIC 变得十分复杂，因为所有的 11 个函数的计算模块都要在芯片上被实施。但这其实对硬件设计者来说，也不过是一个小小的不方便而已。如果有一个针对 X11 的 ASIC 诞生，那么马上会废弃掉 X11 的 CPU 和 GPU 挖矿。

X11 的哈希函数出自何处?

从 2007 年至 2012 年，美国国家标准委员会组织了一个竞赛，这个竞赛选取新的哈希函数家族来作为 SHA-3 的标准，大量包含了设计文档和源代码的哈希函数作为候选方案被提交。虽然有很多候选方案在竞赛中被证实并不符合密码学安全规范，但其中有 24 个哈希函数经受住了所有已知的密码学攻击，X11 选择了其中的 11 种，包括获得最终胜利的 Keccak[①]。

另外被提出但还没有被实施的一个方法是使用一个移动的目标值来作为挖矿解谜。也就是说，解谜算法本身就会变化，就像比特币里的难度会周期性地改变一样。在理想的状态下，为上一个解谜算法而被优化过的挖矿硬件，对下一个解谜算法不再适用。我们不是很清楚要多久改变一次解谜算法，才能达到我们需要的安全要求。如果这是由另类币的开发者所决定的，这可能就变成了一种不可接受的中心化来源。比如，开发者可以根据他们已经开发出来的一种硬件（或者只是优化过的 FPGA），去设计一个相对应新的解谜算法，他们自然就有了针对这个新算法的早期优势。

或许这些解密算法的顺序能够被自动生成，但这看上去也很难。一个想法是选择一大堆哈希函数（比如 24 个没有被攻破的基于 SHA-3 的算法），然后每个用上 6 个月到一年，在这么短的周期里很难开发新的硬件。当然如果这个顺序安排被事先知道，相应的硬件设计就可以按照函数使用的时间表来进行。

ASIC 的蜜月

目前市面上还没有针对 X11 的 ASIC 面世，即便都清楚这种芯片的生产是可能的，这种现象显示了可能很有用的规律。因为适用 X11 算法的另类币的市值都不高，简单来说，还没有足够的市场价值吸引人去设计和生产针对 X11 的 ASIC。一般来说，设计 ASIC 的前期投入都很高（不管是时间还是资金），同时

① Keccak 算法为 SHA-3 的一种加密标准。——译者注

生产单个硬件的利润相对来说比较低。因此，对于新的还没有被证实的加密数字货币，是不值得去投资研发针对性的 ASIC，因为在新的硬件设备可用之前这个货币就可能失败了。即使有一个明显的市场需求，也会有硬件研发生产到出货的延迟。第一批比特币的 ASIC 从最初设计到最终出货花了近一年时间，这在硬件行业里已经算是很快了。

正因为如此，任何使用新的挖矿解谜算法的另类币都会经历一个 ASIC 蜜月期，在这段时间内，用 GPU 和 FGPA 挖矿（或许 CPU 挖掘）的利润会比较高。对于永久阻止 ASIC 的浪潮不太可能，但是吸引个人参与挖矿（并且因此而获得新币）的做法，在新币还处于步步为营的阶段，还是有价值的。

对于抵抗 ASIC 的争论

我们已经可以看到，从长远来讲做到反 ASIC 是不太可能的。但是也有一些其他意见，觉得从已经被证明的 SHA－256 解谜算法转变为一个密码学角度偏弱的新解谜算法，会存在一定的风险。甚者，SHA－256 的挖矿 ASIC 已经接近当今硬件效能的极限了。这意味着，ASIC 所带来的指数型增长可能结束了，之后 SHA－256 挖矿也会因此给网络带来最大的稳定性。

最后，还有一种意见认为，在短期内反 ASIC 也不好。在第 3 章中，我们曾探讨过即使一个拥有全网 51% 算力的矿工，他如若尝试做出很多类型的攻击，也并不理性。因为这样一来会使币值汇率崩溃，使得矿工在挖矿设备上的巨额投资大幅减值，他通过挖掘赚来的比特币的价值也会大幅下降。

对于一个高度反 ASIC 的解谜算法，这个安全性的说辞可能会站不住脚。举例来说，一个攻击者可能会暂时租用巨大算力［比如像亚马逊（Amazon）的 EC2 服务］，用它来攻击，然后不会承受任何财务上的损失，因为他们不需要在攻击后继续租用这个服务。相比较而言，对于一个“对 ASIC 友好”的解谜算法，攻击者就不得不控制一大堆只可以用作加密货币挖矿的 ASIC。这样的一个攻击者其实应该是看好比特币未来的发展，做了一个最大限度的投资。按照这个逻辑进行推理的话，为了最大限度地保护安全，或许挖矿解谜算法应该被设计成不仅仅要让有效的挖矿 ASIC 被设计生产出来，更应该让那些 ASIC 除了用

于加密货币的解谜运算之外，没有任何其他用途！

8.3 有效工作量证明

在第5章，我们讨论了比特币挖矿的能量消耗（有些人会说是浪费）是个潜在问题，经济学家称之为负外部性。我们估计比特币挖矿要消耗几十万千瓦的电能。所以一个明显的问题是，这些用来解谜运算的工作量是否对社会有所贡献？这其实是一个资源再生循环的问题，也会增加社会对加密数字货币的政策支持。当然，这个解谜算法也必须满足几个基本的要求，才能够在一个共识协议里被使用。

以前的分布式计算项目

在比特币诞生好多年之前，就有利用空闲的电脑［或者叫“空闲周期”（spare cycle）］来做一些其他工作的想法。表8.1列出了最受欢迎的几个志愿者运算项目。所有这些项目都有一个特性，使得它们适合成为解谜算法的运算。具体来说，它们需要解决的都是一种“大海捞针”型的问题，可能的答案存在于一个非常大的空间（或者说范围），搜索空间的每一小部分都可以进行并行的快速验证。最有名的例子是在SETI@home网站上，志愿者们被分配一小段无线电信号，用闲置的个人电脑来分析这段信号可能存在的模式以寻找外星文明，同时分布式计算网站（distributed.net）的志愿者被分配一小段可能的私钥来进行验证。[①]

志愿者运算项目，成功地把一个很大的计算任务拆分成小份的任务，然后分配给每一个志愿者进行运算检查。事实上，这种模式在一个特别的叫作伯克利开放式网络计算平台（Berkeley Open Infrastructure for Network Computing，简称BOINC）上是很普遍的，这个平台被开发出来就是用来给不同的个体分发小份

① 大约有500万人参加这个计划，包括译者本人。——译者注

额计算工作的。

在这些应用里，志愿者们主要都是被解决某个问题的兴趣所吸引，即使这些项目通常也会设立一个排行榜来让人们炫耀他们所贡献的算力。排行榜也导致一些人在自己的工作量上作弊，有一些被报告的工作量其实并没有实际完成，这也使得有些项目再分配一些额外的工作去检查网络上的这种作弊行为。金钱，是加密数字货币分布式计算应用的动力，只要技术上是可能的，一定会有参与者尝试去作弊。

表 8.1　热门的志愿者运算项目

项目	成立时间	目标	影响
Great Internet Mersenne Prime Search	1996 年	找到大的梅森质数	连续 12 次发现最大的质数包括 $2^{57885161}-1$
distributed. net	1997 年	密码学的暴力破解演示	首次公开成功地暴力破解了 64 位的密码私钥
SETI@ home	1999 年	寻找外星人	迄今为止最大的分布式计算项目，有 500 万以上的参与者
Folding@ home	2000 年	蛋白质折叠模拟在原子级别上的实现	史上最大算力的志愿者运算项目，超过 118 篇科技论文被发表

有效工作量证明的挑战

有了这些成功的项目，我们可以尝试简单直接地利用这些解决问题的成功方法。例如，在 SETI@ home 的项目中，志愿者们被分配一小段无线电信号监听去寻找外星人，我们可以判断，外星人存在的概率，要比解谜算法找到“获胜”答案并且允许找到答案的矿工去创建一个区块的概率小很多。

但这个想法有几个问题。首先，并不是所找到的答案都有同样的概率成为“获胜”的答案。参与者可能会意识到有特定区域会有更高概率找到异类，那么参与者就会有倾向性，只针对一些能产生不同寻常结果的区域进行分析。对于一个中心化的项目来说，参与者被分配工作，所以所有的区域最终都会被分析

（当然对最有希望的区域会予以优先考量）。对于挖矿来说，任何矿工可以随意尝试任何区块，所以矿工会先涌向最有希望的区块。如果更快的矿工知道他们可以先尝试最有希望的区块，这就意味着解谜算法可能不是一个过程无关的算法。比特币的解谜算法与之相比就有不同，比特币的解谜算法中用来产生一个有效区块的临时随机数都是完全平等的，所以所有矿工都会随意选择一个临时随机数去尝试。这个问题展示了我们之前都已经习以为常的比特币解谜算法的一个主要特征：一个机会均等的解谜区域。

其次，考虑到 SETI@ home 项目中存在着固定的数据量需要被分析的问题，这些数据基于射电望远镜（radio telescope）的观察。随着挖矿算力的不断增长，有可能某一天就没有需要加工的数据了。比特币在这方面也有不同，比特币算法有无限的 SHA－256 解谜可以被创造出来，这就说明了另一个重要的特征需求：永不枯涸的解谜库。

最后，考虑到 SETI@ home 的项目中，有一个受信任的中心化的管理员机构，负责发现新的无线电信号并判断志愿者们应该研究的内容。同样，由于我们使用解谜算法来构建一个共识机制算法，不可能假设一个中心化的机构来管理所有的解谜，这样我们就需要所有解谜的最后一个特征：通过算法自动生成。

哪种志愿者运算项目可能适合解谜算法

回到表 8.1，我们可以清楚地看到，像 SETI@ home 和 Folding@ home 这样的项目不太适合去中心化的共识机制协议，两者都被证明了缺乏我们所列出的上述三个特性。distributed. net 上的暴力破解密码学项目可能适用，虽然它们通常被某些公司用来做某种加密算法的安全评估，但是不能通过算法自动生成。我们可以通过算法自动生成被暴力破解的加密方法，但是某种程度上这就是 SHA－256 不完全原像（partial preimage）算法已经做过的事，并且它没有任何有益的功能。

那就只剩下互联网梅森质数大搜索（Great Internet Mersenne Prime Search，简称 GIMPS）项目了，这个最具备可用性。这个办法的挑战是通过算法自动生成（找到下一个比当前最大质数更大的质数），以及谜底空间是不可穷尽的。事实上，质数的寻找确实是无穷的，因为质数的个数已经被证明是无限个的（特

别是梅森质数是无限量的）。

梅森质数方法的唯一缺点，是需要花费很长的时间来寻找梅森质数，并且梅森质数非常罕见，事实上在过去18年里，梅森质数大搜索项目一共才发现了14个梅森质数，显然在区块链上每年才增加不足一个区块是不可行的。这个问题看起来是缺乏可调节的难度特性，我们在8.1节讨论过这个特性是非常关键的。无论如何，类似于寻找质数这样的解谜算法，看起来是可行的。

质数币

到2015年为止，唯一在实际中被应用的被证明具有有效工作的系统是质数币（Primecoin）。质数币的主要挑战是为质数找到一个“坎宁安链”（Cunningham chain）。坎宁安链是指k个质数的序列P_1，P_2，…，P_k，以使得$P_k = 2P_{i-1} + 1$。也就是说，你选一个质数，然后把这个质数乘以2再加1以得到下一个质数，直到你得到一个和数（非质数）。含有2，5，11，23，47就是一个长度为5的坎宁安链，按照这个规则所获得的第六个数字95并不是质数（95 = 5 × 19）。最长的已知的坎宁安链的长度是19（从79，910，197，721，667，870，187，016，101开始），有一个被推测以及被广泛认可但没有被证明过的理论认为，存在一条任意的长度为k的坎宁安链。

现在，要把这个理论变成一个可计算的解谜算法，我们需要三个关键的参数m、n和k，稍后我们会具体解释。对于给定的一个解谜挑战x（上一个区块的哈希函数值），我们选择x上的前m位数。我们可以认为任何长度为k的链或者大于k的答案是正确的，这条链上的第一个质数是一个n位质数并且和x一样有m位的首段数据（n≥m）。值得注意的是，我们可以调整n和k的值，来让这个解谜变得更加困难。增加k的值（需要的链的长度）使得问题难度指数型增长，而增加n的值（链上的第一个质数的长度）使得问题难度线性增长，这就可以让我们对问题难度进行微调。其中，m的值只需要足够大，使得在知道前一个区块的值之前的预先计算方法变得没有意义。

其他我们所讨论的属性看起来已经都有了：结果可以很快被校验，问题本身是无关过程的，题库可以无限大（假设对质数分布的知名数学推导是正确

的），然后解谜可以通过算法做到自动生成。实际上，这个解谜算法已经被质数币用了两年，并且对许多给定的 k 值产生了坎宁安链里最大的质数。质数币还做了进一步的扩展，在其工作量证明中涵盖了其他类似的质数链，包括“第二”坎宁安链，其中$P_i = 2P_i - 1$。

这验证了在某些限定的情况下，有效工作量证明是具有实际运用的。当然，寻找大的坎宁安链有用与否，是有争议的。坎宁安链当然也代表了我们已知数学知识宝库的一小部分，其在未来可能会有一些应用场景，但在目前还没有实际的应用出现。

永久币和存储量证明

另外一种有效工作量证明叫存储量证明（proof of storage），也被称为可恢复性证明（proof of retrievabitlity）。不同于需要一个单独计算的解谜算法，我们可以设计一个需要存储大量数据被运算的解谜算法，如果这个数据是有用的，那么矿工在挖矿硬件设备上的投资就可以被用于大范围分布式存储和归档系统。

让我们看一下永久币（Permacoin），这是第一个用于共识机制的存储量证明方案。首先我们讨论一个大文件 F，我们假设所有人都认可 F 的价值并且这个文件不会被改变。例如，当一个加密数字货币上线时，由一个可信任的分发者选择 F，这有点类似于任何一个加密数字货币启动时都需要一个创世区块，理想状况下这个文件会具备公共价值。例如，大型强子碰撞（Large Hadron Collider，简称 LHC）的实验数据，这个数据已经达到了几百拍字节（petabytes，用 PB 表示）的大小，对这些数据的备份是很有价值的。

当然，因为 F 存储量非常巨大，大多数参与者都无法对整个文件进行存储，但我们已经知道，在不需要了解整个文件的情况下，如何使用密码学里的哈希函数来确保每个人都对 F 认可。最简单的方法是，每个人都认可 H（F），但更好的方法是用一个大型梅克尔树来代表 F，所有的参与者都认可梅克尔树的根值。现在，每个人都认可 F 的价值，证明 F 的任意一部分是正确的就变得很有效率。

在永久币系统中，每一个矿工 M 存储着任意 F 文件的子集 $F_M \subseteq F$。为了实

现这一点，当矿工产生一个公钥 K_M 来接受资金时候，他们就对该公钥进行哈希运算以生成一个区块 F_M 的虚拟随机数集，他们必须存储这个数集以实现挖矿的目的。这个子集就会变成某个固定数量的区块 k_1 的一部分，我们必须在这里做一个假设，当矿工开始挖矿的时候，他们有办法获得这些区块——可能是从一个标准文件源地址下载下来（见图 8.2）。

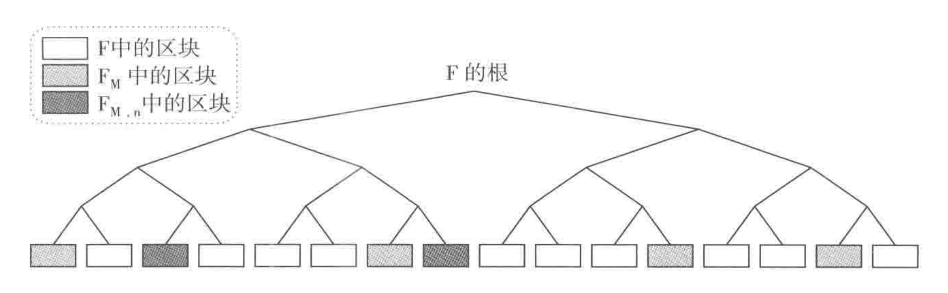


图 8.2　在永久币系统中选择一个文件的随机区块

注：在这个案例中，$k_1=6$，$k_2=2$。在实际应用中，这些参数会大很多。

一旦矿工在本地存储了 F_M，这个解谜算法就非常类似于传统的 SHA－256 挖矿了。给定前一个区块的哈希值 x 时，旷工选择一个临时随机数 n，将其进行哈希运算并产生一个虚拟随机数子集$F_{M,n} \subseteq F_M$，这个子集包含了 $k_2<k_1$ 个区块。值得注意的是，这个子集是由所选的临时随机数和矿工的公钥共同产生的。最后，矿工对 n 以及 F_k 中的区块，进行 SHA－256 的哈希函数运算，如果计算的结果是低于目标难度的，那么也就意味着他们找到了一个有效的方案。

校验一个解谜算法的结果需要以下几个步骤：

- 校验 $F_{M,n}$是由矿工的公钥 K_M 和临时随机数 n 共同产生的。
- 通过检验其在梅克尔树节点到全局统一的树根路径，来检验 $F_{M,n}$中的每一个区块是正确的。
- 校验 H（$F_{M,n} \| n$）的值比目标难度要小。

我们很容易看出，为什么解谜过程需要矿工在本地存储所有的 $F_{M,n}$。对于每一个临时随机数，矿工都需要计算 $F_{M,n}$中随机子集的哈希值，如果通过远程

访问一个存储空间来获取文件，就会非常慢，几乎不可能实行。

不同于 Scrypt 算法的案例，如果 k_2 足够大，并没有一种可行的类似于时间内存的权衡方案。如果矿工仅仅在本地存储了一半的 F_M，并且 $k_2 = 20$，那么在他们找到一个不需要从网络中取回任何文件区块的临时随机数之前，他们必须要尝试 100 万次，降低一定量的存储负担会以计算量指数型增长为代价。当然，由于 k_2 梅克尔树路径要在所有的路径中被传输和校验，如果 k_2 设得太大，也会使运算变得非常低效。

k_1 的设定也可以有所权衡。更小的 k_1 意味着矿工需要更少的本地存储空间，因此这种挖矿就更加民主化（更多的人可以参与）。然而，这也意味着，大量的矿工即使有能力提供更大的存储空间，他们也没有动力去存储多于 k_1 个 F 区块。

同样，这是一个对完整的永久币做了细微简化的方案，但是对我们理解整个设计的关键部分来说是足够的了。最大的应用挑战，当然是找到一个合适的大文件，这个文件要有一定的重要意义，同时也是公共的，需要保存多个备份。如果 F 文件本身随着时间的推移会发生变化，或者随着时间的变化而调整难度，这样会使方案变得更加复杂。

长期的挑战和经济意义

总结一下本节内容，有效工作量证明是一个非常自然的目标。考虑到一个好的共识机制所需要的其他解谜算法，实行起来也有相当大的挑战。即使如此，至少本文所举的两个案例——质数币和永久币——在技术上是可行的，虽然它们也都有一些技术方面的缺陷（主要都是需要更长的时间去验证解谜结果）。此外，对比在比特币挖矿中动辄数百万美元的投入以及大量电力的消耗，这两种加密数字货币的应用都对社会公益有一些贡献。

有效工作量证明是否应该是纯公益的，有一个有趣的经济学方面的争议。在经济学中，公益的意思是非排他性的，也就是说所有人都可以参与使用，并且是非竞争性的，对公益的其他用途不应该影响其本身的价值。一个经典的例子就是灯塔。

我们这里所讨论的案例，比如蛋白质折叠（protein folding）[①]，就不是一个纯公益的项目，因为有一些公司（比如大的制药公司）可以从中获利。实质上，这些机构挖矿的成本会相对变低，因为它们可以获取其他人无法获得的额外利益。

8.4 不能外包的解谜算法

我们现在再看一下对于替代挖矿解谜的另一个设计重点：防止矿池的产生。我们在先前的第5章里谈到，大部分的比特币矿工都会加入一个矿池，而不是独立挖矿。这就造成了少量矿池拥有绝大部分挖矿算力的现象。由于每个矿池都有一个中心化的管理方，有些人担心这其实违反了比特币去中心化的核心设计原则，会危害到比特币的安全性。

拥有大部分算力的矿池显然是一个问题，任何一个中心化管理的矿池可能会实施一套自定义的挖矿策略，然后用它来攻击网络。这种矿池也是黑客们攻击的目标，因为通过攻击矿池可以迅速地控制大量的挖矿算力。矿池管理员也可能会删改交易或是强迫收取更高的交易费。矿池中拥有大多数矿工，意味着大部分矿工都没有运行一个完全有效节点。

有意思的是，这些担忧有着现实世界的影子，比如选票。在美国和其他许多国家，出售选票是非法的。加入一个被一方控制的矿池，和在比特币的共识协议里出售你的选票有点类似。

矿池的技术要求

回忆起来，矿池看起来是一个突然发生的现象。并没有证据显示，中本聪在比特币的最初设计中考虑过矿池的概念。在互相不信任的个体之间运行一个

---

① 蛋白质折叠问题被列为“21世纪的生物物理学”的重要课题，它是分子生物学中心法则尚未解决的一个重大生物学问题。——译者注

有效率的矿池，这样的事情在最初的几年里看起来不太现实。

正如我们在第 5 章所看到的，矿池通常会指定一个管理员，他有一个大家都知道的公钥。每一个加入的矿工还是按照往常一样进行挖矿，然后递交“近似”或者“部分”答案给矿池管理员，这些答案在低级别难度的时候可能就是一个有效答案，通过这种做法来证明他们做了多少工作量。当矿池中的某一个参与者找到了一个有效区块的时候，这个管理员会按照每个人所提交的工作量的占比来分配奖励。虽然有很多种不同的分配方式，但是所有矿池都遵循这个基本模式。

正因为如此，矿池的存在依赖于比特币的两大技术特征。第一，一个矿工很容易通过提交工分来证明（概率上）他所做的工作量。不管实际上找到一个有效区块是多么困难，通过设定一个足够低的合格工分的临界值，矿工可以容易地证明他们在任意精度的工作量。考虑到我们需要解谜题目可以在任意难度上被创造出来，这个问题看起来很难改变。

第二，矿池成员可以容易地向管理员证明，他们遵守规则并且通过实际运算来寻找有效区块，然后矿池会作为一个整体接受奖励。这是行得通的，因为这个矿池的公钥是被写进币基交易，并包括在区块里的梅克尔树上。即使一个矿工找到了一个有效区块，甚至只是一个近似区块（也叫工分），他也无法改变整个矿池的公钥，而成为新铸币的接受者。

“区块丢弃”攻击（block-discarding attack）

矿池的这种设计有一个弱点：没有办法来确保矿工在找到有效区块的时候一定会提交给管理员。假设有一个矿池成员对一个大型矿池不满，他可以正常地参与挖矿然后提交工分，但他在找到一个有效区块（可以让整个矿池获得奖励）的时候，并没有告诉管理员而是直接把它丢弃掉。

这个攻击降低了整个矿池的挖矿能力，因为攻击者的工作量并没有实际贡献到挖矿中去。但是这个矿工依然会收到奖励，因为他看起来也在不断地提交工分，只是运气不好没有找到有效的区块。如果这个矿池的奖励设计方案是收入中性的（也就是所有的挖矿奖励都被分发到每个参与者），那样的话这个攻击

会让这个矿池亏损。

这种攻击被称作民间攻击或者是蓄意破坏攻击，这也被认为是一种蓄意破坏，因为这个攻击看上去对攻击者和矿池都是不经济的、代价不菲的。这个攻击者本身也会遭受损失，因为他所丢弃的有效区块将会使他放弃他应该有的一部分奖励回报。当然，这个攻击者还是会由于其他一些挖矿解密算法而获利。

看起来一个理性的矿工不会采用这种策略，因为他会有所损失而不会得到任何实际的回报。但（令人惊讶的是）在某些情况下，这个策略是可以有利可图的，我们在下文有所讨论。但是无论如何，我们想要设计一个全新的挖矿解谜算法，以确保这种策略永远都是有利可图的（以抵抗矿池的存在）。

矿池之间的区块丢弃攻击

好多年以来，人们都觉得进行区块丢弃攻击是无利可图的，实际上如果两个矿池之间的互相攻击却不一样。这种方案已经被提出来好多次，伊泰·艾瑞尔（Ittay Eyal）2015年的论文中首次深入分析了这种攻击模式。

我们考虑一个简单的案例：假设两个矿池A和B，每个有50%的全部挖矿算力。现在假设B动用了一半的能力（25%的总体算力）来加入矿池A挖矿，然后把所有找到的有效区块丢弃掉。我们可以推演，在一个简单的模型里，B会赢得5/9的所有奖励，大于他正常挖矿时候所获得的50%的奖励。在这个简单的案例里，动用一半的挖矿算力去攻击矿池A对矿池B来说是一个最佳的策略。

这个案例随着矿池数量的增加而变得更加复杂。截至本书撰写之时，丢弃区块攻击在实际中还没有被大范围观察到。但长期来看可能性还是存在的，像这类攻击会对大型矿池的运营产生关键影响。

奖励破坏

我们设计这种攻击的目的，是让矿工们即使加入了一个矿池挖矿，也会缺

乏向矿池管理员提交有效区块的动力。目前，只有矿池管理员可以获取挖矿奖励，因为管理员要求所有的参与者在他们挖矿的币基交易中加入一把特殊的公钥。这个公钥是否被正确地放入，可以在提交近似区块的时候被很容易地检查验证。矿池管理员是唯一知道私钥的人，因此可以决定新铸币的走向。

但如果我们要求所有的参与者都知道私钥（这样一来，当找到有效区块的时候大家都可以重新定义区块奖励的去向）呢？为了做到这一点，我们需要一个解谜算法，每一个解谜运算的尝试都要求知道币基交易里的私钥。我们可以把解谜从“找到一个区块，其哈希值低于一个特定的目标”改成“找到一个区块，这个区块里的数字签名的哈希值低于一个特定的目标”。这个数字签名必须要用币基交易里同一把公钥来计算。

这样的解谜算法，会给矿池管理员两个都不可靠的选择：他们可以把私钥分发给所有成员，如此，他们之中任何一人都可以私自挪用全部矿池资金。另外一个办法是他们可以代表矿池成员进行签名。计算一个签名的计算量比计算一个哈希函数要大许多，这样一来，矿池管理员会承担主要的苦活与累活，所以最好让矿池管理员成为一个独立的矿工。

不能被外包的挖矿的优劣

由于这类解谜算法不能够有效地（并不是完全不可能）被外包到一个不能被信任的参与者，这就使得成立一个由不被信任的参与者所组成的矿池变得十分困难。它可以有效地阻止所有的矿池形成，即便是像 P2Pool 这样成立一个没有矿池管理员的去中心化矿池。

存在如下争议，部署这类解谜算法可能会不可抑制地造成更多的中心化，而不是更少。因为概率上较高幅度波动（找到有效区块而获得奖励的概率问题）会让小矿工们不敢参与挖矿，剩下的只会是大型挖矿团队。目前，虽然矿池表面上控制了大量的挖矿算力，但还是不清楚如果他们想利用这个优势来发起攻击的话，其中许多成员是否会叛逃。大型挖矿矿池和可以承受高幅度收入波动的小矿池，到底哪个风险更大？这是一个未能解决的问题。

设计一个共识协议，理想方案是小额度地奖励每个找到低等难度解谜答案

的矿工，以“自然地”降低概率波动风险。这就意味着矿工们不需要组成矿池，同时小矿工们还可以参与挖矿获利。仅仅降低每个区块产生之间的时间间隔不会起到作用——它需要被降低1 000倍或者更多，才能够在概率风险上与大型挖矿矿池所面临的情况相当。但到那个时候，每个区块之间的间隔只有不到一秒，陈旧区块的数量会变得不可控制的高。还有一个问题，是否存在另一种共识协议，可以做到在不需要瞬时广播所有解谜结果的情况下，让解谜运算变得更加容易？

8.5 权益证明和虚拟挖矿

在结束本章之前，我们讨论一下这个想法：用虚拟挖矿（virtual mining）来替代计算力挖矿。虚拟挖矿是指一组不同的挖矿方法但它们都有一个共同的特点——对参与的矿工只要求少量的计算资源。

建立一个封闭挖矿系统

作为一个思想实验，假设比特币或是其他加密数字货币成为全球主要支付手段。矿工起初会拥有一些加密数字货币来购买挖矿设备和支付电耗，以此获取一些新币来作为挖矿的奖励（见图8.3）。这基本上是个消耗资源的过程。

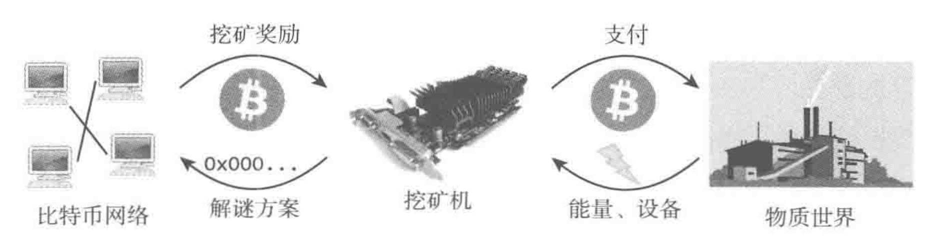


图8.3 比特币挖矿的资源循环

一旦挖矿设备变成了一种商品，并且电力也是一种商品（本来就是），没有矿工有任何优势可以更有效地把他们起初拥有的加密数字货币转化成挖矿奖励。

除非有细微的效能差距，挖矿投入最多的矿工将会获得最大的挖矿奖励。

推动虚拟挖矿的基本问题是：如果我们把挖矿设备和能耗这一环节省去，会产生什么结果？毕竟，这个过程主要是用来证明谁在挖矿这件事情上投入最多。为什么不简单地把挖矿“算力”按比例分配给当前所有的持币人？

回忆一下比特币挖矿的初衷是在区块链上建立起一个投票机制，有更多算力的矿工会得到更多的投票权力。因此，我们可以设计一个“投票”系统，选票（投票权力）是由每个人所拥有的当前币量所决定的。

虚拟挖矿的优势

这个方法的优势是显而易见的：它把如图 8.3 中右边浪费资源的一半去掉了，留下了一个封闭的系统，如图 8.4 所示。

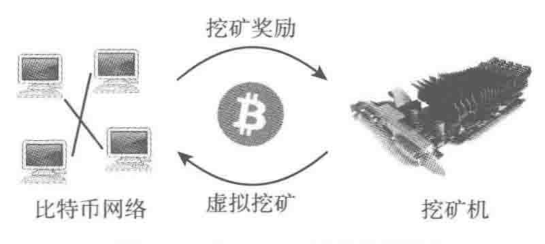


图 8.4　虚拟挖矿的资源循环

除了简单化之外，这个方法会大大减少比特币对环境的影响。它不会把能耗降到零，因为矿工总是会消耗一些计算资源来和网络通信验证，有一些虚拟挖矿方案也要求少量的挖矿计算力。但总体上，比特币里绝大部分的挖矿工作量可以被省去。

虚拟挖矿还可以阻止中心化的发展趋势。因为没有硬件，所以也不必担忧有 ASIC 的问题。每一个矿工挖矿的效率都和其他人完全一样。任何虚拟矿机所用的解谜算法都是反 ASIC 的。

这可能是虚拟挖矿最重要的一点。虚拟挖矿可能解决了我们在讨论反 ASIC 解谜算法时候所遇到的问题，也就是考虑到货币的长期健康，矿工可能不会去投资挖矿设备的生产。任何一个比特币的持有人其实也是这个货币的利益相关

者，一个强大的虚拟矿工（比如持有51%或更多的币）是一个非常大的利益相关者。他们有原动力来做对整个系统有利的事，因为这样一来他们所持有的币也会增值。这比“矿工已经投入了大量的挖矿设备，且设备价值会基于未来的币值，所以没有人会进行恶意行为”的说法更有力。

这就是“权益证明”这个名字的来源。除了节省挖矿设备和节省能源之外，可能虚拟挖矿的最大动力，来自这个货币的利益相关者有着强烈的意愿成为这个系统的维护者。

实施虚拟挖矿：点点币

有许多种不同的虚拟挖矿，我们在这里只讨论最常见的几种。我们要强调的是，这些想法还没有被严格地用科学的方法研究过，也没有像比特币的工作量证明一样因为比特币的普遍性而经过实战洗礼。

我们先看一下在2012年启动的**点点币**（Peercoin），是第一个使用权益证明的另类币。点点币是工作量证明与权益证明的一种混合体，“拥有量”以“币龄”为计价单位。一个特定的还没有被使用交易的输出的币龄，是“这个输出里的币量”与“这个输出里还没有被使用过的区块数量”的乘积。现在，为了挖到一个区块，点点币的矿工也必须像比特币的矿工一样去进行一个SHA-256的解谜运算。但是，这个解谜运算的难度会随着他们想消耗多少币龄调整，消耗越少难度就越低。为了做到这一点，这个区块包括一个特殊的“币拥有量交易”（coinstake transaction），在这个交易里，有些交易被用掉只是为了把它们的币龄重设成零。这些在币拥有量交易中被消耗的币龄总和，决定了工作量证明解谜运算中发现一个有效区块的难度。

矿工可以在最初用很大的计算力和一些很少的拥有量来挖矿，但是可以用公式来设定难度：当一些币龄被消耗后，找到有效区块会变得十分容易。这个运算型解谜的效果主要是为了保证，在有两个矿工尝试消耗同样大小币龄的情形下，这个过程仍然是随机的。

许多其他的虚拟挖矿另类币方案使用了略微不同的设计，包括NxT、BitShares、BlackCoin和Reddcoin。在这些设计方案里，一定数量的币被消耗用于使

运算型解谜变得极为简单，这使得解谜运算不再是挖矿过程中最主要的挑战。

权益的其他形式

有两种混合模式值得探讨：

- **权益证明**。最简单的权益证明模式是使那些拥有大量币控制权的矿工挖矿更加容易。这类似于点点币的币量和币龄混合证明，只是在这个模式中不考虑币龄。这个模式的劣处是，不像点点币那样每次成功获得有效区块之后重置币龄，最有钱的参与者总是可以最容易地挖矿。
- **储量证明**。在这个模式里，当一个矿工用一些货币来铸造一个区块的时候，这些货币针对一定数量的区块被冻结。这可以被想象成币龄的一个镜像：这个系统奖励那些希望在未来一段时间不消费的矿工，而不是那些在过去一段时间内不使用货币的矿工。在上述两种情况下，矿工的收入来自因为不能使用货币去做其他事情的机会成本。

无利害关系问题

虚拟挖矿是科研的前沿领域，还有许多未解的问题。即使有一些加密数字货币已经启动并且使用了虚拟挖矿，它们都面临和比特币一样的压力，即防御有目的性的攻击者。

虚拟挖矿有一个常见的漏洞，被称为“无利害关系问题”（nothing-at-stake problem）或者“股权粉碎攻击”（stake-grinding attacks）。假设一个有着小于50%的币拥有量的攻击者，尝试制造一个有k个区块的分叉，如同我们在前面讨论过的，这样的分叉攻击有着相当高的失败概率。在传统挖矿里，一个失败的攻击有着很高的机会成本，因为矿工本可以在挖掘的过程中赚得奖励，而不是浪费挖掘资源在失败的攻击上。

但在虚拟挖矿里，这个机会成本并不存在。一个矿工可以既在当前最长的区块链上挖矿，同时又可以进行一个创建分叉的尝试。如果分叉成功，则会消耗掉大量的筹码；如果失败，这个失败的记录不会出现在最长的区块链上。因

此，理性的矿工也会不断地尝试分叉攻击。

对于这个问题，有一些不同的解决办法。大多数的虚拟挖矿方案都积极地使用检查点来防御长分叉攻击。但是正如之前讨论过的，这有点和去中心化的共识协议概念背道而驰了。

分叉攻击和检查点

当你下载比特币核心钱包软件时，有几个硬编码的检查点，或者过去区块的拼接。这样做的根本目的是让首次下载区块链更加顺利。如果没有检查点，其他节点可以使用伪造的（但有效的）区块和分支来冲击你。对于当今的攻击者来说，在低区块高度产生有效的谜题解决方案是非常简单的，那就是接近起源区块，因为初始阶段的难度相对较低。你最终会发现这些区块不在最长的有效分支上（更精确地来说，不在最高总体难度的有效分支上），但你必须浪费资源来做这件事情。

有些另类币，特别是虚拟挖矿计划，已经采取了以检查点作为防御分支攻击的强大形态。节点会从指定检查节点收到检查点的常规更新，该更新由指定的私钥签发。节点会放弃与检查点冲突的分支。这种机制使得检查点的运作方，尤其是另类币的创建者，能从分叉和“转回”区块中选择胜出者。这种设计非常有趣，但是已不是去中心化一致认可的协议。

以太坊（Ethereum，一个在2015年启动的另类币，我们将在第10章中详细讨论），建议了一个称为“Slasher”的方法来惩罚尝试进行分叉攻击的矿工。在Slasher方案中，使用筹码去挖矿需要用私钥对当前区块进行签名，来应对那些进行作弊的交易，如果矿工曾经使用相同的筹码去签署两个不连续的区块链（不是前后关系），Slasher允许其他矿工可以在区块链上输入这两个签名作为作弊的证据，并且拿走一部分筹码以作为奖励。虽然看起来这个方案非常有效，但是协议本身非常复杂，还没有被实际部署。

一个终极的防御攻击方式可能存在，就如同我们在传统挖矿方案中看到的，

矿工可以简单地没有足够的动力去进行攻击，因为即使攻击成功，也会危害整个系统并使得他们所拥有的筹码贬值。

虚拟挖矿的其他弱点

虚拟挖矿还有其他两个弱点值得提及。第一，在某种形式的虚拟挖矿方案中，即使“股权粉碎攻击”不存在，也可能使得某些类型的攻击变得容易，因为挖矿“蓄力”（save up）是可能的。例如，大量的币可以被积蓄起来，直到可以进行一次剧烈的挖矿变化使得分叉成为可能。即使是某个类似于 Slasher 这样严禁同时在两个区块链上挖矿的系统上，也是可能的。为了防止这样的攻击，点点币限制了币龄参数不能超过 90 天。

第二，如果虚拟挖矿中的某个矿工获得了 51% 的筹码，他可以通过只在他的区块上挖矿的方式永远保持这个优势，基本上也就意味着可以控制整个区块链。如果有新的筹码和交易费从区块奖励中产生，那个拥有 51% 的矿工也会抢去这些奖励，这会让他的筹码慢慢接近 100%。在传统挖矿模式中，即使有一个 51% 的矿工存在，永远可能存在拥有更大算力和更低能耗的其他矿工出现，并且会减少最大矿工的市场份额。在虚拟挖矿里，很难避免这个问题。

虚拟挖矿有可能真的成功吗

在比特币的主流社区里，虚拟挖矿是有争议的。有一个说法是，系统的安全性必须建立在真正的资源消耗上，也就是动用真正的电脑硬件和消耗电能去进行解谜运算。如果这个理论成立，工作量证明上的能源耗费可以被看成是系统的安全费用。但这个论点还没有被证明，就像虚拟挖矿的安全性也没有被证明一样。

总结来说，人们想改变比特币挖矿解谜算法的很多方面，这也是研究与创新的重点区域。到目前为止，还没有一个替代方案具备理论健全性和实用性。例如，即使 Scrypt 算法在另类币中很受欢迎，但是也没有做到真正的反 ASIC，而且其用途也还不清楚。当然，替代的解谜算法完全有可能在未来获得更大的成功。毕竟，比特币本身也经历了数十年的不断失败的尝试与发展，才最终成

为一个既有很好的设计理念又有相当的实用性的加密数字货币。

延伸阅读

定义刚性内存功能和建议范本的论文是：

Percival, Colin. "Stronger Key Derivation via Sequential Memory-Hard Functions," 2009.

您可以通过如下网址阅读：

https://www.bsdcan.org/2009/schedule/attachments/87_scrypt.pdf.

关于内存范围（memory-bound）功能的早期论文包括：

Abadi, Martin, Mike Burrows, Mark Manasse, and Ted Wobber. "Moderately Hard, Memory-Bound Functions." *ACM Transactions on Internet Technology* 5 (2), 2005.

Dwork, Cynthia, Andrew Goldberg, and Moni Naor. "On Memory-Bound Functions for Fighting Spam." In *Advances in Cryptology—Crypto 2003.* Berlin: Springer, 2003.

关于 Cuckoo Cycle proposal 的研究包括：

Tromp, John. "Cuckoo Cycle: A Memory-Hard Proof-of-Work System." IACR Cryptology ePrint Archive, 2014.

您可以通过以下网址阅读：

https://eprint.iacr.org/2014/059.pdf.

关于永久币的介绍请参阅：

Miller, Andrew, Ari Juels, Elaine Shi, Bryan Parno, and Justin Katz. "Permacoin: Repurposing Bitcoin Work for Data Preservation." In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, 2014.

您可以通过如下网址阅读：

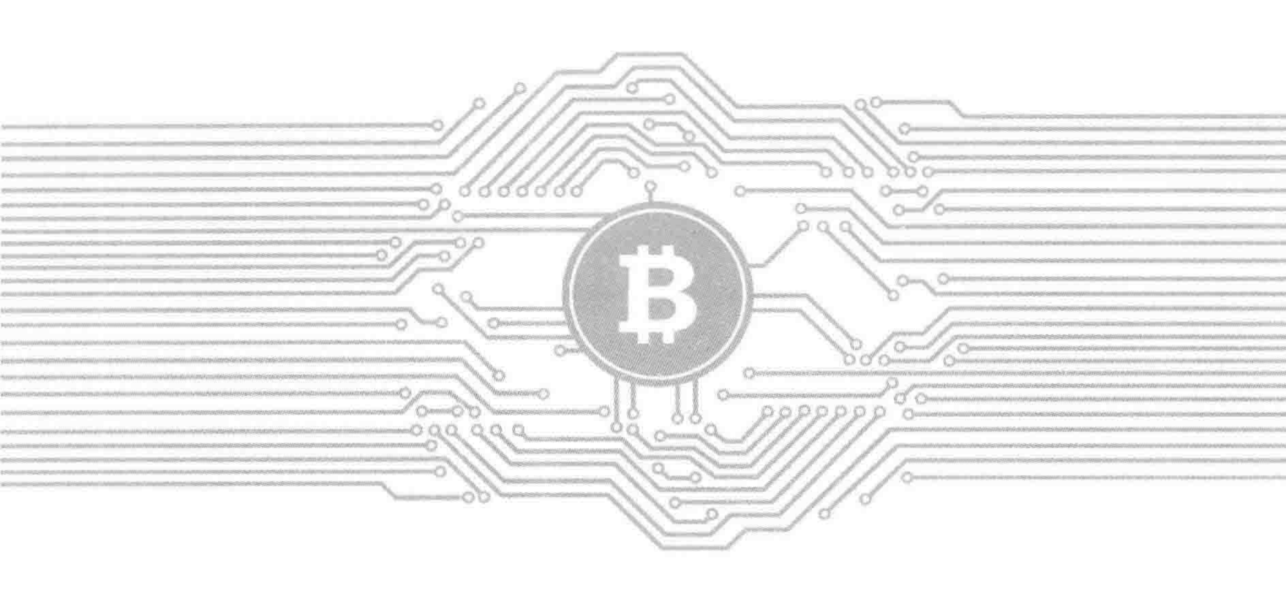
http：//research. microsoft. com/pubs/217984/permacoin. pdf.

讨论不同的哈希函数和 SHA -3 竞争的论文是：

Preneel, Bart. “The First 30 Years of Cryptographic Hash Functions and the NIST SHA-3 Competition.” In *Topics in Cryptology—CT-RSA, 2010*. Berlin: Springer, 2010.

关于不可外包谜题的介绍是：

Miller, Andrew, Elaine Shi, Ahmed Kosba, and Jonathan Katz. “Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions.” In *Proceedings of the 22nd ACM Conference on Computer and Communications Security*, forthcoming.



第9章

比特币“平台”



在前面的章节里，我们展示了比特币的技术基础架构和它作为货币的机理。现在我们来探讨除了货币之外，比特币可以作为核心组成部分的其他应用。这些应用，有些是直接利用了比特币的当前特性，并未做任何修改，而更多的则是需要做一些微小调整的。

我们根据应用的实用性和学术上的趣味性选择了下述应用案例，尽管案例并不能穷举，但这些关于比特币运作的例子（或者设想中的运作模式），将给您对比特币新的用途看法带来一些启示。

9.1 比特币作为一个只能被添加的记录

比特币是一个只能添加而不能删除的记录。它是一种可以不断添加新的数据，但是数据一旦被添加上去，就变得不可修改并且永久保存的数据结构。因此，通过比特币，我们可以获得一个时间顺序：判断一个数据是在另一个数据之前还是之后被写进了记录的。这个次序是由区块之间的哈希函数指针，而不是区块上的时间戳所决定的，因为时间戳可以作假，或者是由于矿工更改时间戳的值使其变得更小（更早），或者是矿工的计算机时钟没有同步，更或者是由于网络延时产生的差异。话虽如此，如果一个区块的时间戳延迟了好几个小时，它就会被其他的矿工们拒绝。所以时间戳还是相对准确的。通过下面的示例，我们可以看到，这些特性是有实际用途的。

安全时间戳

比特币这种只能被添加的记录特性可以被用来建立一个**安全时间戳**（secure timestamping）系统。假如，想要证明在时间 T 我们就知道了 x 的值，但并不想披露它的具体值。只有在未来很长时间后，当有可能需要证明我们确实知道这个值的时候，才有可能需要去披露它（当然，如果我们在时间 T 知道 x 的值，我们在 T 之后的时间还是知道这个 x 的值）。而且我们一旦证明了这一点，就需要使这个证据具备永久性。

在第 1 章中我们看到，可以用哈希函数来锁定数据 x。我们不需要公布 x 值本身，取而代之地，只需要在区块链里公布这个数据值的哈希函数 H（x），即可以来证明我们知道这个 X 值。这个哈希函数的特性，保证了我们不可能再找到另外一个数据 y，其哈希函数结果与 x 的哈希函数结果一致，也就是说，当 y≠x 时，H（x）=H（y）是不存在的。我们还可以依赖哈希函数另外一个常用的特性：只要 x 本身具备比较高的**最小信息熵**分布特性（distribution with high min-entropy），也就是说，x 是不可预测的，那么 x 的哈希函数结果不会透露关于 x 的任何信息。如果 x 本身没有这种不可预测的特性，就像我们在第 1 章中探讨的，我们可以选择一个有较高的最小信息熵分布的随机数 r 和 x 组合签名，然后用 H（r | x）作为对外公布的一个数值约定。

这个设想的核心就是，我们在时间点 T 只向外公布哈希函数 H（r | x），然后在之后的某个时间点来公布 r 和 x。任何人看到这个只能做增量的记录，都会相信在我们发表 H（r | x）的时候我们一定知道 x，因为没有除此之外的其他方法可以让我们产生那些数据。

时间戳的应用

这个安全时间戳到底有什么用途呢？一个可能的应用就是可以用来证明创意的优先性。假设我们想证明，申请专利的一些创意点子早就在我们头脑里存在。我们可以在产生创意的第一时间，就将设计文档或者示意图草稿在区块链里用哈希函数发表出来，但不会向任何人泄露这个创意的具体内容。之后，当

我们提交专利申请或是公布这个想法的时候，我们可以将最初的设计文档和相关信息发表出来，任何人都可以查证这些文件的函数时间戳，来证实我们在这之前，也就是我们发布设计文档的哈希函数约定的时候，就已经有了这个创意（证明我们对这个创意的时间上优先所有权）。

我们还可以用同样的方法来证明，其他人收到过我们发给他们的信息。假设爱丽丝雇用鲍勃去做一个编程的工作——他们之间的合同规定，鲍勃必须在一定的时间内将他所做的工作内容提交给爱丽丝。双方都想要获得一个保证，如果将来对相关工作内容有争议，比如鲍勃是否按时提交程序或者他提交的内容是否满足合同要求，双方都希望可以有相关的事实去证明。为了确保这一点，他们可以互相协商，将鲍勃提交的工作内容共同签名后，再在区块链上发表它的哈希函数。任何一方如果对提交的时间或者内容撒谎，另一方可以通过披露哈希函数的输入来重现当时发表的内容，来证明对方是错的（比如在法庭上）。

安全时间戳功能还有许多其他有意思的应用。有一种完全公钥签名方案，被称为盖伊·福克斯签名方案（Guy Fawkes signature scheme），相比较通常的公钥签名，只是利用了哈希函数和只能做增量记录的记录特性，而不需要任何重量级的加密算法。

对未来预测证明的攻击

我们目前无法仅仅用时间戳就证明对未来的准确预测，能对未来进行预测（clairooyance），当然非常好。而且从表面上看，这好像可以做到。在一个事件发生之前（比如一个体育比赛或是选举）发表一个对结果的预言，然后在事件发生之后，再证明在之前就已经预测到了。但这个方法是否真正可行？

在 2014 年下半年的世界杯决赛阶段，有人想用这个办法来“证明”世界杯的组织方国际足联（Federation Internationale de Football Association，简称 FIFA）在搞腐败。当时有个推特账号，由于在一些重要比赛之前就可以准确地预测到这些比赛的结果，因而被广受关注。比如，该账号准确地预测到德国队会在加时赛取胜并且马里奥·格策（Mario Götze）会进球。这看起来可以证明，要么

是微博主人有预知未来的能力，要么是比赛被操纵了。然而事实上，这个博主只是在比赛前发布了所有可能发生的事件，比如，对于所有的参赛球员，都有一条关于他会进球的微博，以及对于每一种可能的最终比分，都有一条相关的微博等（见图9.1），然后在比赛结束之前，博主删除了所有那些不准确的预测，只留下那些准确的“预测”。

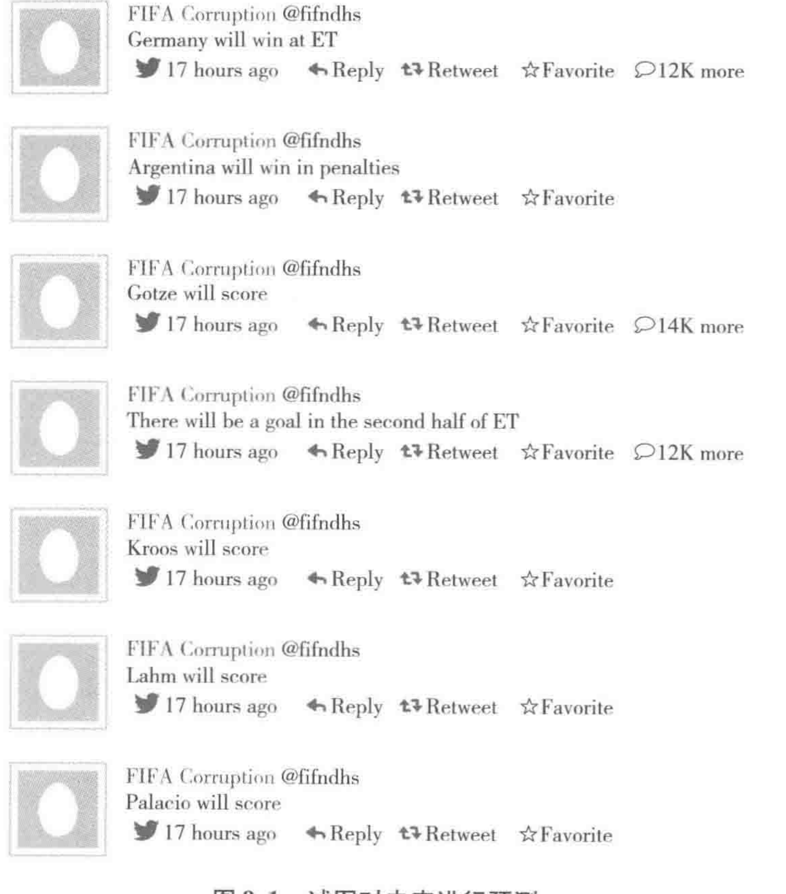


图9.1 试图对未来进行预测

注：这就是那个虚假的试图通过预测比赛结果来“证明”世界杯决赛圈的比赛被操控了的推特账号。其中第一个和第四个在赛后被证明是正确的，其他不准的预测就被删掉了。

可以用同样基础的攻击方法攻破任何安全时间戳系统。你只需要在事先预埋下所有的结果，然后最终只披露那个正确的结果。这就意味着如果你想证明

你有预测能力，就必须去证明你做且只做了一个预测结果，而不是多个预测。但如果你想基于哈希函数揭示结果，是很难实现的，尤其是在比特币的区块链上，因为安全时间戳系统并不将承诺与任何个人身份识别相关联。如果你不揭示它们，就会很容易公布很多种承诺，而那些你从未揭示的承诺很难轻易追溯到你。

过时的安全时间戳

这里介绍一个简单的低科技含量的安全时间戳方案：通过刊登广告，你可以在一份报纸或者其他媒体上登出你预测结果的哈希函数值，相关的旧报纸杂志会被保存在图书馆里或者在线备份。这种方法可以提供较高程度的保证，证明你在报纸发出的当天就已经知道这个结果了。以后，当你想要披露你预测的结果时，你可以在同一个报纸上刊登第二份公告。

比特币里的安全时间戳

如果我们想用比特币而不是报纸来实现时间戳的功能，我们应该在哪里放置约定的哈希值？是在交易中的某个环节，还是直接在一个区块里？

人们想出来的第一个也是最简单的解决办法是，直接把钱打到数据的哈希函数值，而不是公共钥匙的函数值。由于你不知道对应地址的私钥，这样做会“消耗”这些币，让它们销毁掉，并且永不能被利用。为了降低成本，你可能需要发送微量的币值，比如1聪（satoshi，0.0000001个比特币，这是比特币的最小交易额）。

这个方法虽然很简单，但消耗比特币的做法不讨人喜欢（即使和交易费相比，这种被消耗的比特币量级可以忽略不计）。更大的问题是，因为比特币矿工不知道这些交易开支是永远不可用的，他们会永远地追踪下去。因此整个比特币社区对这个方法都不太感冒。

另一个较为先进的被称为**承诺币**（CommitCoin）的方法，是将你的数据编码进私钥里。第1章中曾经提到过：“使用ECDSA时，确保随机性良好来源至关重要，因为不良来源将可能导致密钥信息的泄露。这一点不难理解，如果你

使用了不良随机源来生成密钥，那么该密钥就可能不安全。但是 ECDSA 的古怪就在于，即使你仅仅只是在生成签名时使用了不良随机源，而你使用的密钥完美无缺，你的个人密钥还是可能会被泄露。”

承诺币利用了这个特性。我们生成一个新的私钥把我们的数据约定进行编码，并对应地生成一个公钥。然后我们会发送一个微小金额的交易（比如 2 000 聪）到那个地址，随后再发送两笔每次 1 000 聪的交易回来。最重要的是，当发送回来的时候，我们会用同样的随机源来对两次交易进行签名。这样，任何人在区块链里计算包含被封装的数据约定的私钥时，必须使用两个签名。

比起把数据约定编码到公钥的方法，承诺币避免了消耗额外的比特币，而且矿工不再会一直追踪一个永久不能再被使用的支出。不过这个方法十分复杂。

不能被再次使用的输出

一直到 2015 年，比特币实行时间戳的办法是用一个 OP_ RETURN 的交易，这个交易的输出可以被证明，但不能被二次使用（见图 9. 2）。这个 OP_ RETURN 指令会立刻返回一个错误代码让这个脚本永远不能成功地执行，这样一来，所封装的数据就被忽略了。就像我们在第 3 章看到的，这既可以用做消耗证明，也可以用来编码任意数据。到 2015 年，OP_ RETURN 允许输出 80 个字节的数据，这对哈希函数来说是足够了（SHA - 256 需要 32 个字节）。

OP_ RETURN < H （data）>

图 9. 2　用 OP_ RETURN 指令的时间戳

注：这是一个“不能被再次使用”的交易输出脚本，中间封装了一个数据约定。

这个方法“挤出了”在没有被使用过交易支出里的“水分”，因为矿工会精简 OP_ RETURN 里的支出。这个数据约定的花费其实就是一个交易费。在整个 2015 年，一个典型的交易费通常小于 1 美分。这个交易费可以分摊在针对多个数据的一个约定上，从而使得成本更低。在 2015 年年末，已经有几个网站在做这些服务。它们收集不同用户的一组数据，把这些数据封装到一个梅克尔树中，然后发布一个包含了这个梅克尔树树根数据中不能被再次使用的交易支出。

这种做法就好比，把当天需要实行时间戳的所有用户数据封装到了一个数据约定里。

非法内容

区块链随意封装数据的特性也有不好的地方，可能会被某些人恶意使用。在大多数国家，有些内容，尤其如儿童色情，它的制作和传播都是非法的，并且会伴随非常严厉的处罚。著作权法也严格规定了某些内容的传播。

当然，不少人已经尝试这样做去“危害”或者扰乱比特币社区。比如，有报道称有部分色情链接被公布在比特币的区块链上。这些害群之马的目的，就是让下载比特币区块链到个人硬盘并且运行完全有效节点的行为变得很危险，这也意味着你有可能存储和传播了这些非法的信息。

然而，截至目前，还没有好的办法来阻止这种写入任意数据到比特币区块链的行为，即使我们用 P2SH（支付给脚本的哈希值）来防止恶意攻击行为，也只不过是使交易多花些费用而已，无法完全阻止这种行为。

好在法律不是计算机算法，尝试用技术的手段对法律进行“黑客攻击”虽然很诱人，但并不容易。法律是需要人类来解释的，并融合了其他因素，比如我们的意图。以美国联邦法案 2252 号为例，其中在描述有关拥有、分发传播和接收儿童色情制品的非法行为时，使用的措辞就用了“明知故犯”这样包含了意图的关键词。

另外一个值得注意的是，根据上面我们讨论过的字节大小的限制，图片数据（除非是非常小的图片）不能直接被写在区块链的数据块中，这些数据要么被存放在只在区块链中保存相应链接的外部数据库中，要么是用一种冗长的办法封装在多个交易之中。最终的结果就是，大多数比特币用户都没有能力在交易中直接解码并查看数据，更不用说解码并查看跨越多个交易的数据了。

依附在比特币上的附着币

从好的一方面来说，因为我们可以把任何数据都写进比特币的区块，从而在比特币的系统之上建立起一个全新的货币系统，而不需要开发一个新的

共识机制。我们只需要简单地把比特币用作一个只能被添加的记录，然后把我们开发新币所需要的所有数据写进比特币的区块链。我们称这种方法为一个依附在比特币上的“**附着币**”（overlay currencies）。比特币成为一个底层基础架构，所有附着币的数据，通过以不可消费的交易支出的方式写进比特币的区块链。

当然，比特币的矿工不会验证你写进区块链的数据，因为他们不知道也并不关心这些数据在你所定义的新的货币体系里是否正当有效。只要你肯付交易费，任何人都可以写任何东西。不同的是，你必须自己开发更加复杂的逻辑来验证新货币体系里的交易，然后在每个收发这种新币的客户端（也就是钱包软件）都必须有这套逻辑。

举例来说，一个附着币的矿工不能再拒绝双重支付的交易。相反，每个附着币的用户必须检查区块链里的历史记录。如果有人尝试重复支付这个币（已经被用过一次了），那样第二次的交易就应该被直接忽略。因为这个缘故，在附着币里没有一个轻量级的 SPV 客户端。

合约币（Counterparty）是其中一种比较优秀的附着币，所有合约币的交易都被写入比特币的区块链，在 2014 年，大约有 0.5% ~1% 的比特币交易携带了合约币的数据。同时它支持的功能也比比特币更多、更丰富，因为合约币不需要开发新的共识机制，而比特币的矿工也不需要了解合约币的规则，合约币的开发者可以集中精力开发一些有趣的功能，比如智能合约、用户自定义货币等。合约币的 API 也比比特币的 API 丰富很多，因为比特币的矿工不需要理解或者是批准这些 API 的开发。

不需要开发新的共识机制就可以创造一个新的数字货币，这种可能性是十分诱人的。你甚至不需要去鼓励新矿工们来加入你的系统，也不需要去改变比特币就可以增加新的功能特性。但是，这种系统还是依赖于比特币的，比如，这些附着币的交易费规则就受制于比特币。另外，由于附着币上的节点可能需要处理大量的数据，而比特币不会帮你去过滤这种交易，这种方法也有可能是低效率的。

9.2 比特币作为一个“智能资产”

我们现在来探讨一下，除了货币功能，比特币平台的其他特性。

我们在前面第6章中谈到，你可以简单地通过跟踪交易图谱，就可以在比特币系统里追踪一个币的所有权。请记住这一点：没有一个具体意义上的比特“币”，只有未消费的支出，我们把它们叫作币。每个比特币都有一个历史记录，任何人都可以在区块链里查询到。一个币的历史记录可以追溯到一个或多个原始交易，这些原始交易标志着这个比特币的诞生。正如我们之前讨论过的，在比特币里，匿名性其实是个伪命题，因为你可以通过这个方法去追踪比特币的所有权。

可互换性（fungibility）

比特币的这个特征让我们发现了它的一个有趣的现象：比特币不是可互换的。在经济学中，一个具备可替代性的商品是指所有的个体是相同的，然后可以互相替换。比如黄金就是可以互换的，一盎司纯金可以和另一盎司纯金互换（因为它们之间没有任何差别）。但是比特币不一样，每个比特币都是独一无二的，因为每一个比特币都有着自己独特的历史记录。

在很多场景下，不同的历史记录可能不会有什么差异，但是如果特定的历史记录对某些人比较有意义，那么在你和他们交易的时候，你的一个比特币和他们的一个比特币就不一样。可能有些人不愿意用他的比特币来和你交换，可能因为他更喜欢他的比特币的历史记录，例如，部分重视旧币价值的收藏家们，可能觉得从创世区块里造出的币有着特殊的价值。

智能资产

比特币的这个可追溯性特性有什么作用吗？我们已经看到它可能会危害比

特币的匿名性。接下来，我们要看一下为什么比特币的历史记录会有意义。

让我们先思考一下，怎样让一个普通的线下的物理货币有意义？假设我们想要在物理货币中加载一个元数据，事实上已经有人在这么做了。例如，在纸币上涂些文字，通常是一个笑话或者是一种“政治宣言”。但这么做纯粹为了好玩，并不影响纸币的价值。

但如果我们可以把证实过的元数据“黏”在我们的货币上，而这些元数据不是轻易就可以复制的，又会有什么结果呢？有一个做法就是把加密签名包含在元数据内，然后把这个元数据和钞票上的序列号进行绑定。

但这又有什么用呢？比如一个棒球队，如果想用纸币作为门票，那么采用这个做法，他们就不需要花费大量精力去印制门票，也不用担心有人会去伪造门票。纽约扬基队可以宣称一张有特殊序列号的美钞可以作为一场特殊比赛的入场券，并且指定到某个特定的观赛席。这些特殊的纸币可以采用与其他门票同样的方式分发，比如邮寄给在线购买球票的球迷。任何拥有这张特殊纸币的人，都可以凭此进入体育馆，并坐在指定的座位上观看比赛。这张纸币本身就是门票。

扬基队可以用数字签名来增加真实性。他们可以把特定的比赛日期、座位号及钞票的序列号一起做签名，然后把这个签名印在纸币上。通过一个简单的二维码就可以实现这个功能（如图9.3所示）。球馆可以相应地维护一个保存所有钞票序列号对应每场比赛和座位号的数据库，当你凭票入场的时候，它们只需要根据你所提供的二维码去数据库里校验即可，也就不需要在纸币上盖章并印上相关信息了。

但这样做究竟有什么好处呢？现在，纸币可以代表许多事物。上述例子中，纸币替代了体育比赛门票，除此之外，纸币还可以有其他许多应用。为了纸币不能被伪造，政府投入巨大，我们可以利用纸币上已具备的防伪特性，来创建其他应用。当然，这张纸币的本身价值也保存了下来。当一个球迷使用了这张门票后，这张纸币还可以正常流通。当然，如果每个人都想在钞票上印一个元数据可能会有问题，但我们可以用数据库的方法来规避这个问题。

当然，这个新的元数据是否有意义，完全取决于我们对数据发行者的信任。



图 9.3　一张普通的钞票上设置一些有用的元数据

在上面这个例子中，一定有人知道存在一个特定的“密钥”来签发有效的扬基队球票，或者下载整个扬基队的数据库以识别这个特殊纸币的门票价值，而对其他人来说，这就是一张普通的一美元纸币。无论如何，这是一个不错的属性，因为一旦在这张“门票”完成使命之后，它又可以作为普通纸币进入货币流通。

染色币

在比特币上，我们是否可以采用类似的数字化的方式增加元数据呢？我们想保留比特币好的特性，比如可以在线交易、快速结算，以及不依赖于银行。

顾名思义，**染色币**（Colored Coins）就是把比特币“染色”，即使这个币几经倒手，我们也可以根据这个特殊的“颜色”来追踪比特币，就如同在物理货币上印上一个代表特殊数据元的图章一样。一个“染色”的比特币依然可以作为一个有效的币，只是携带了额外的元数据。

为了做到这一点，在一个被称为“发行”的交易里，我们嵌入一些额外的元数据来宣布某些比特币具备了特定的颜色。如图 9.4 所示，在一个交易的支出中，我们发行了 5 个“浅灰色”的比特币，同一交易支出里的其他的 7 个，仍然是普通的没有染色的比特币。另外一个人，可能持有一把不同的签名密钥，在其他的交易里发行了“深灰色”的比特币。我们称之为“染色”，是为了便于直观理解。在实际中，所谓的“颜色”其实就是一串二进制的数字代码。这里，最重要的一点特性是，同样颜色和同样价值的币是完全相同的。

虽然我们现在有不同颜色的比特币，但依然可以进行正常的比特币交易。

我们可能碰到一个交易，它包含了几个不同输入的比特币：有些是深灰色的，有些是浅灰色的，有些是没有染色的，并且混在一起。同时，这个交易可能会有几个支出交易，其中的比特币保持着被染色的状态，并且交易中可以添加一些元数据，决定这些比特币根据染色的不同去往不同的交易支出，我们可以把一个包含 4 个深灰色币的支出拆分成两个更小的深灰色币组合，我们也可以把几个深灰色币组合到一个大的深灰色币交易中。

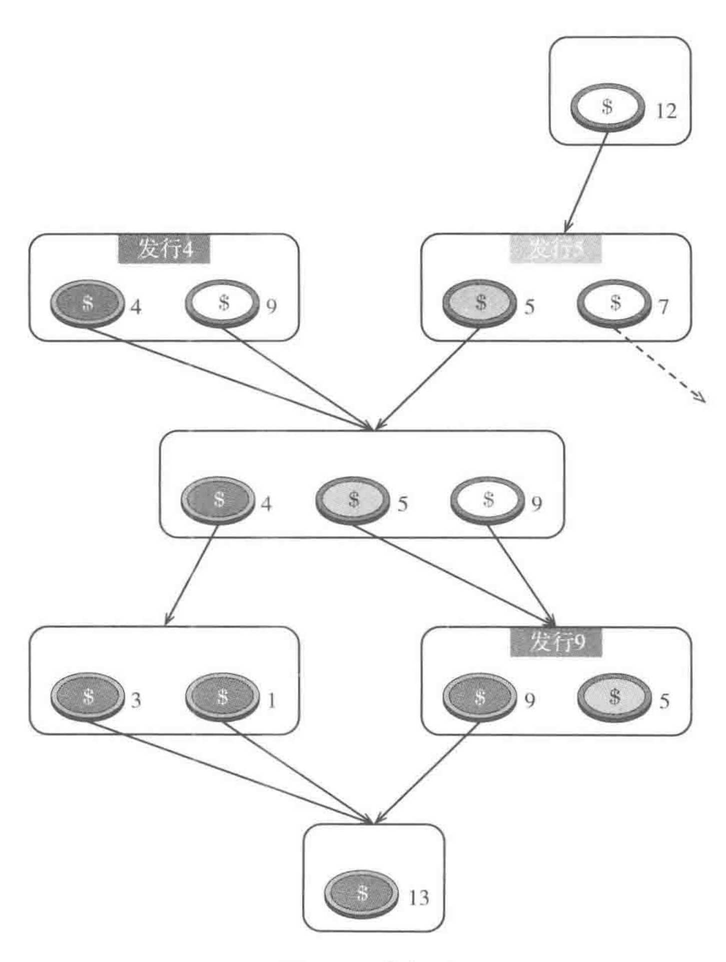


图 9.4　染色币

注：交易图谱描述了染色币的发行和传播过程。

开放资产

2015年，在比特币里实施“染色”的、最受欢迎的应用方案是“**开放资产**”（OpenAseet）。资产通过一个特殊的支付给脚本的哈希值（P2SH）的地址来发行。如果你想发行染色币，首先要选择一个P2SH地址。任何通过这个地址转账的币在进入的时候都没有颜色，在出来的时候，就会被这个地址分配一个特定的颜色。如果要让这个染色币变得有意义，你还要把这个地址公布出来。有许多交易所会追踪这些地址来推断比特币染上何种颜色。由于比特币可以按照时间顺序通过多个地址进行染色，因此，一个比特币被染上多个不同的颜色，也就不足为奇了。

一旦执行了一个带有染色币的交易，你就必须嵌入一个有特殊标记的支出，就像我们给数据约定加时间戳一样，这是一个可证实且不可再次消费的支出。被封装在这个有特殊标记的支出里的元数据，详细地列出了染色的输入是如何分配到不同输出的。

正如我们之前所注意到的，比特币对此完全兼容，因为它并不对比特币做任何改变，矿工社区也没有对这种做法多加干涉。无须中央权威授权，它允许任何人对货币进行各种染色。只要有人理解、认可并遵循你所设计的染色币的规则，那么你所发行的染色币甚至有可能超过比特币本身的价值。比如，如果扬基队发行染色币，并且这些币可以作为球场入场门票，只要球场的管理员认可这些染色币的门票特性，他们就会让你凭票（染色币）入场。

这个方法的第一个缺点是，我们必须把不可被再次消费的标记支出放进每个交易里。由于每次交易一个染色币时，都需要多花一些交易费，这会增加一点成本。第二个缺点是，矿工只会验证作为底层基础的比特币的有效性，但不会去验证染色币的有效性。如果想要验证一个染色币的有效性，你必须亲自去查证它所有的历史记录，或是委托第三方来帮你完成。此外，不同于一般的比特币，用户不能使用一个轻量级的简单支付验证客户端。这就使得类似手机这种有计算性能限制的设备，在使用染色币时变得非常困难。

染色币的用途和智能资产

一个经常被引用的智能资产用途就是公司股票。一个想要用染色币来发行股票的公司，需要公布一个发行地址和规则，通过这个地址发行的染色比特币代表了公司的股票。1 聪比特币就可能代表公司的 1 股。股票持有者就可以在区块链上交易股票，而不需要一个像股票交易所这样的中央媒介。当然，股票持有者必须信任公司对这种股权的认可。例如，公司承诺按照每股支付相应的股息或者授权股东对公司决策有比例性的投票权利。传统股票是通过法律规定来保障的，截至 2015 年，染色币或者其他基于区块链的资产还没有获得任何一个司法机构的认可。[①]

物理特性。另一个可能的用途是，染色币可以代表现实世界中的一些资产。比如，一个染色币可以代表一处房产或者一辆汽车。你所拥有的一辆高级轿车可以和一个在区块链上的特定的染色币关联，然后持有这个染色币的人可以用它来启动和驾驶这辆车。当你要卖这辆车的时候，或是至少要转让使用权的时候，你只要在区块链里简单地执行一个单笔交易。在第 11 章，我们将探讨在技术上如何实现这个功能，以及一些可能会遇到的社会和法律上的障碍。但是，由于其具备了智能特性，我们的梦想是让现实世界中任何有形资产都可以用染色币来代表，并且资产的转让或交易就像比特币的转让或交易一样容易。

域名。最后一个例子是，使用染色币来完成一些现有域名系统的功能：登记和转让拥有权，关联域名和 IP 地址。域名市场有许多有意思的特征，其中之一是它有无数可能的域名。根据域名名字是否方便记忆以及其他因素，域名有着不同的价值；同样，对于不同的人，一个域名可能有完全不同的意义。我们可以用染色币来处理域名登记和其他相关功能。事实上，有一个另类币就是为这个目的而设计的，叫作**域名币**（Namecoin），我们会在第 10 章中做具体讨论。

---

① 2016 年 3 月，美国在线零售商 Overstock 已经获得了美国证券交易委员会（SEC）的批准，通过比特币区块链技术发行 100 万股新股。——译者注

每个方法都有自己的独特优势，染色币可以获得比特币区块链的安全保护；而另类币可以相对容易地部署域名登记、转让、关联 IP 等复杂逻辑。

9.3 多方参与的安全博彩系统

我们现在来讨论一下如何在比特币里举行一个“掷硬币”的游戏。首先看一下，在线下是如何建立这个系统的。

爱丽丝和鲍勃想要下一个 5 美元的赌注。他们在下注之前商量好了游戏规则。鲍勃往空中扔一个硬币，爱丽丝在硬币落地之前叫出“正”或“反”面。当硬币落地的时候，可以立即判断谁是赢家。双方都知道这个结果有足够的随机性，他们之中任何一人都没有办法影响结果。

为了使双方相信这个游戏是公平的，游戏的步骤顺序以及硬币的特性至关重要。但上述设计有一个缺陷，就是他们二人都必须同时在场，而且相信对方会愿赌服输。在线上，我们也想设计一个同样“公平”的博彩系统，同时确保输家也会愿赌服输。

初看起来，这个应用有些古怪，并且有局限性，并不值得深入研究。非常有意思的是，一个基于比特币的在线博彩系统中本聪之骰，已经被证明非常受欢迎，但它并未采用上述设计模式，而是依赖于某一方的信用，但它时不时地囊括了大部分比特币网络上的交易量。

我们想研究这种加密数字货币的“掷硬币”系统的真正原因是，如果我们可以据此设计一个安全协议的话，也可以用这个技术来设计其他有趣和有用的协议。密码学专家研究“多方参与的安全计算”，也就是说多个互相不信任的参与者，每个主体都有各自的数据，然后综合各主体的数据来共同计算一个结果，但同时每个主体都不想让其他参与者知道自己的数据是什么。想象一个类似的场景，一次竞价拍卖，但没有一个可靠的拍卖行。通常这些计算需要被随机化，来打破互相之间的关联，最后，这个计算的结果是有金融属性的，并且是不可逆转的。比如，我们想要保证中标者最后会付款给拍卖物品的卖方，更进一步，

让卖方的（智能）资产自动转移到中标者的名下，甚至更进一步，我们还想要惩罚那些不守规矩的人。

总而言之，一个安全的多方博彩系统虽然看起来简单，但其实可以用来研究一个非常强大的系统模式：各自都有敏感数据的互不信任的一群参与者，共同来执行一个程序，不仅仅是为了控制数据，还可以控制与之关联的资金。

在线掷硬币系统

第一个挑战是找到与“掷硬币”相对应的线上的相关方法。假设我们有三个参与者：爱丽丝、鲍勃和卡罗尔，大家都想以相同的概率来选择一个号码 1、2、3。我们尝试以下协议：每个人选择一个大的随机数，比如爱丽丝选 x、鲍勃选 y、卡罗尔选 z，然后互相告知各自的随机数，并共同计算结果 (x + y + z)%3。

如果他们都是完全独立地选择随机数的话，这个方法是可行的。但请记住，这是在互联网上，没有办法可以限制他们绝对地“同时”送出数据。爱丽丝可能会等到鲍勃和卡罗尔送出随机数之后再发布她的数据。这样一来，她可以轻易地操纵这个计算的结果。我们没有办法设计出一个数据交换协议，它可以让大家都相信没有人会作弊。

为了解决这个问题，我们还是要回到函数约定。首先，每一个人选一个大的随机数，并发布它的哈希函数值；然后，每个人披露各自所选的数字；接着，其他两个人查证这个被披露的函数值和在第一步发表的数据是否正确；最后，计算这个三个随机数的结果，如下：

> 第一回合：
>
> 每个参与者选择一个大的任意字符串。爱丽丝选 x，鲍勃选 y，卡罗尔选 z。
>
> 每个参与者发布对应的哈希函数值 H（x）、H（y）、H（z）。
>
> 每个参与者验证 H（x）、H（y）、H（z）具有明显的差异性（否则放弃这个协议）。
>
> 第二回合：
>
> 三个参与者分别披露他们所选的字符串 x、y 和 z。

每个参与者查证这些字符串是否与第一回合里发布的函数值相吻合。

最后的输出是 $(x+y+z)\%3$。

这种数据协议之所以能成功是因为以下因素：第一，因为函数的输入 x，y，z 是大的任意数，没有人可以在第一回合之后预测其他人的输入；第二，如果爱丽丝按照规则任意地选择她的输入，她可以相信，不管鲍勃和卡罗尔是否选择了随机数，最后的输出结果也是随机的。

公平性

要是有人不披露约定怎么办？在这个协议的第二回合里，假设卡罗尔一直等到爱丽丝和鲍勃披露他们的秘密随机数，然后，在披露之前，意识到她会输掉这一局，她就有可能拒绝公布她的随机数——她可以说她忘记了或是假装下线。爱丽丝和鲍勃可能会怀疑，但他们也没有什么好的办法去追查。

我们所要做的是立下一个规矩：参与者若是做出承诺，则必须在一定的时间内披露所选的随机数。在密码学里，这个特性叫作公平性。比特币提供了一个非常好的解决方法。

比如爱丽丝想要做出一个有时限的约定承诺，但鲍勃是唯一对此有顾虑的。第一，爱丽丝先设置一定的保证金，比特币支出交易的脚本可以用来规定，这笔保证金只能用以下两种支付情形：第一种支付情形是必须同时有爱丽丝和鲍勃两人的共同签名；第二种支付情形是只要爱丽丝披露了她的随机数，以后消费这笔交易，就只需要有爱丽丝的签名。如果爱丽丝所选择的随机字符串是 x，那么输出脚本（ScriptPubkey）会包括哈希函数 H（x）的值。

接下来，爱丽丝和鲍勃会同时签下一个交易，把这个保证金支付给鲍勃（两种情形之一）。但为什么爱丽丝会同意这样做呢？这个交易带有一个 nLock-Time 值来保障鲍勃不能在时间点 t 之前来赎取保证金。因为，在此时间之前，爱丽丝只要愿意披露她的约定随机值，她就可以赎回这个保证金，所以她的这个签名交易是安全的。见图 9.5。

如果爱丽丝在弃局之前没有披露她的约定随机数值，那么鲍勃就可以在时

```
scriptPubKey:
        OP_IF
              <AlicePubKey> OP_CHECKSIGVERIFY <BobPubKey> OP_CHECKSIG
        OP_ELSE
              <AlicePubKey> OP_CHECKSIGVERIFY OP_HASH <H(x)> OP_EQUAL
        OP_ENDIF

scriptSig for Case 1:
        <BobSignature> <AliceSignature> 0
scriptSig for Case 2:
        x <AliceSignature> 1
```

图 9.5　在有时限的哈希函数约定中使用输出脚本和输入脚本的交易输出

间点 t 之后赎取该保证金。没有人逼迫爱丽丝披露她的随机数，但如果她不披露，她会因此失去预设的保证金。

我们如何用这个有时限的函数约定来实现安全的博彩系统？其实，架构和之前几乎一样，差别在于，我们不再采用简单的函数约定，而是采用有时限的函数约定。任何一方，要赎回这笔保证金就必须把正确的随机数值 x 披露出来；如果在最后期限到来时还不披露出他的随机数值，就会放弃他的保证金，以用来补偿其他两个玩家。

可以在比特币系统上实施这个博彩系统。但这个系统有些复杂，而且有时限的函数约定还要求多个非标准的交易。当系统里有 n 个玩家的时候，由于每个玩家都要设置一笔保证金，我们需要 n^2 个约定，此外玩家们还不得不投入比全部赌注更多的资金用来托管保证金。但对于参与者较少的游戏来说，这是合理的，并且有更好的效率。最重要的是，这个游戏验证了本来认为不可能的数据交换协议，比如在线掷虚拟硬币，并对不遵守规则的玩家进行惩罚，在比特币的世界里是可以做到的。

9.4　比特币作为一个公共的随机源

在上一节，我们展示了一群人如何选择一个公平的随机数。在这一节里，我们将讨论如何用比特币来产生一个对任何人都公平的公共随机数。为什么我

们需要一个公共随机数？让我们先看一下几个现实中已经存在的依赖公共随机数的案例。

NBA 新人选秀

其中一个例子就是每年春天的美国 NBA 联赛的新人选秀大乐透。NBA 联盟中所有的30 支球队聚集在一起，根据上赛季每个球队的赛季排名增加相应的权重，随机选择球队选秀顺序。在 1985 年，联盟首次采取这种方法进行选秀，乐透选秀的过程通过电视现场直播，包含球队名字的信封在一个透明的转盘里被充分打乱，委员会专员随后去挑选这些信封。因为纽约尼克斯（Knicks）队获得了当年的状元秀中锋帕特里克·尤因（Patrick Ewing），最终尤因也确实成为 NBA 名人堂的一员，当时这个乐透的产生引起了不小的争议。由于那次的乐透发生在纽约，其他球队的球迷宣称整个过程被人操纵，并偏向尼克斯队。

有很多有关 NBA 是如何操纵选秀过程的阴谋论，比如著名的“折角”论（“bent corner” theory）是说包含有尼克斯的信封有一个角被故意折弯了，这样委员会专员通过触摸就可以分辨出哪个信封是尼克斯的。另外一个论调是说尼克斯的信封之前被放在了冷冻室里，这样专员可以通过选择一个手感比较冷的信封来挑选出尼克斯队。这些论调都反映了一个事实，这种类型的选择要做到绝对公平是非常困难的，有很多合乎推理的作弊空间，想象一下一个职业魔术师的巧手可以做些什么！直到今天，选秀乐透每年都会举行，但每一次都充斥着各种阴谋论和谣言，以说明选秀并不是绝对公平的。

美国军队选秀

另一个更加严肃的案例是 1969 年的美国征兵选秀，用于决定哪些年轻人会去参军，大部分被选中的人事后都被派去参加了越南战争。同样使用了一个类似于 NBA 选秀的方法，由美国国会派出的代表来主持选秀并通过电视直播（如图 9.6）。他们在一个大的塑料桶中放了很多小球，每个小球包含了一个数字，然后轮流从桶中把这些数字小球取出来，根据数字所代表的生日来决定优先级，



图 9.6　1969 年（越战）军队选秀

根据这个优先级挑选合格的年轻人参军。

1969 年的这次征兵选秀，是首次在全国范围内采用乐透方式进行选秀。其目的是通过避开数以千计的本地征兵委员会，以使选拔过程更加公平，并且向公众公开这个过程。但遗憾的是，这个选秀乐透也演砸了，不到一个星期，概率专家通过数据调查分析注意到了一个特别的模式（如图 9.7 所示）：出生于下半年的人被选中的优先级较低。虽然这种差异非常细微，但是从概率统计上是非常显著的，说明这不太可能是偶然事件。当他们回看现场录像的时候，发现每次转动转盘的次数恰恰都是偶数次，这意味着一开始是上层的小球有较大的概率一直留在上层，说明为了形成随机抽签的混合程序并不充分。

这两个案例都证明了，设计一个公众认可的随机过程并由此产生一个认可的公共随机数，是十分困难的。无论你采用什么方法，总有人怀疑你作弊。风险在于：这个随机过程可能会被操作——即便这个过程是真正随机的，但公众并不信任它。①

① 所以一个公平的不受操纵的公共随机源是一个公共福祉，而比特币可以做到这一点，因为它是去中心化的。——译者注

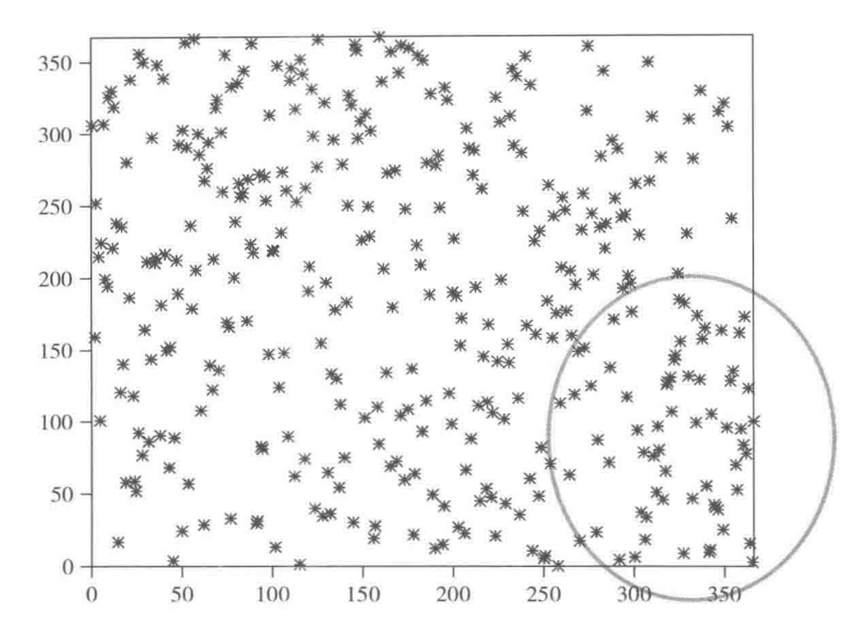


图 9.7　1969 年征兵选秀的概率统计偏差

注：x 轴代表日期，y 轴代表选秀号码。

密码学“信号塔”

由于成本低，并且大众易于理解，用转盘、抛硬币、掷骰子等方法去展示公开随机性，在历史上一直比较普遍。但是这些方法非常难以审计，因此并不适用于大范围的场景。即使整个过程从视频上看起来是合法的，人们也有理由去怀疑乐透的执行者可以使用“魔术师之手”去操纵结果。

那么，通过密码学的办法，我们是否可以做得更好？这里，我们用密码学“信号塔”（cryptographic beacons）来特指一个提供公共随机源的服务。我们的想法是“信号塔”会源源不断地在一个固定的频率产生随机数，并且没有人可以预测这些随机数。只要大家同意这一点——没人可以预测这个信号塔的下一个输出，那么大家就都可以相信其生成的是一个真正的随机数。

一个完美的密码学信号塔可以服务于各种公开乐透项目，比如上面两个例

子。如果你想要在本地的俱乐部玩一个宾果（Bingo）游戏①，你再也不需要用一个大的转盘来产生随机性了。只要每个人都信任信号塔，就不要用物理的方法来展示随机性了，这会省去很多麻烦。

密码学专家提出了很多其他的公共随机源的应用方案，包括投票系统、零知识验证、分割选择协议等。如果你有一个完美的密码学信号塔，其中很多方案都会变得非常简单有效。但遗憾的是，截至目前，我们还没有找到一个完美的方案去打造这样一个信号塔。

NIST 信号塔

NIST，即美国国家标准与技术研究所（National Institute of Standard and Technology）。从 2011 年开始，NIST 运行了一个它们自己的信号塔服务。它们声称用了一个非常复杂的实验室装置来产生随机数，甚至动用了两个纠缠态光子。由于随机数是由量子力学现象产生的，那么理论上这个数字可以保证非常强的随机性。如果你认可海森堡不确定性原理（Heisenberg uncertainty principle）和其他一些被广泛接受的物理学原理，那么你就会相信这个信号塔产生的数是真正随机的，并且不可预测。NIST 信号塔服务可以每 60 秒产生一个附带有数据签名的随机数，并提供一个非常便利的程序接口——服务可以通过网页来访问并返回随机数。

从某种意义上说，NIST 信号塔代表了从物理上展示随机性的极限，但无法解决一个基本的信任问题——你必须信任 NIST 确实是通过它们所宣称的这些程序来产生随机数的，你必须信任在马里兰州的某一个建筑里面，NIST 确实用它们的实验室来产生这些随机数，而不是伪造的，你还必须信任它们确实没有能力故意重写其中一些随机数。

打造一个信号塔的其他潜在方法：自然现象

我们是否可以使用一些每个人都可以观测到的自然现象来实现信号塔？

① Bingo 是一种填写格子的游戏，在游戏中第一个成功者以喊“Bingo”表示取胜而得名。——译者注

或者我们可以使用天气的一些细节，比如在某天某个特定地点的温度，或者是风力的强度，或者是否会下雨。当然，我们有能力去提前预测天气，但是预测的结果并不是非常精确，所以我们可以使用这些测量值的"最低有效位"。但是这个方法也有局限性，那就是需要所有的参与者在同一个地点做同样的测量。

为了避免这个问题，我们可以利用太阳黑子，一种太阳表面的爆发活动(见图9.8)。另外一个例子是宇宙背景辐射，通过广播天线，你可以从地球上的任意一点都可以监测到这个数据，而且每个人都可以读取到同样的数值。这些都是超大范围发生的现象，很容易向公众证明没有其他人可以操纵这个过程。想象一下，某人穿越宇宙到达太阳表面，然后用某种办法去影响太阳黑子现象，而其目的仅仅是为了操纵地球上的某个乐透项目，这显然是不现实的。所以上述这些方法都有很好的特性：公众可观测性、可防止被操纵性，以及一个可接受的不可预测性。

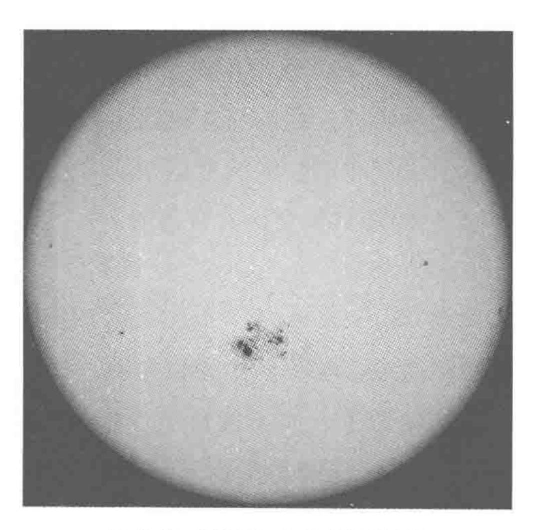


图9.8 NASA太阳黑子照片

但上述这些方法都有一个共同的问题，就是生成随机数太慢了。例如，如果随机信号是当日最高温度，那么你每天只能获取一次这样的数值；太阳表面也不是经常变化，在很多密码学应用中，随机数通常作为伪随机数发生器

（Pseudo-Random Generator，简称 PRG）的输入，从安全角度考虑，这些输入通常需要达到 80 比特位长度，甚至更多，但如果以天气或者天文学为数据源，那我们可能需要很长时间才能积累到 80 位长度的随机数。

另外，观测太阳黑子现象这种方法还需要专业技能，所以你还需要依赖一些可信任的专业观测者来获取这些测量数据。由于有很多这样的可信任的观测者，我们也希望他们“互相之间有诚信”。信号塔的应用者，或者说是这些应用的用户，可以随时选择并替换观测者。这个特性被称为“信任的敏捷性”，相比于 NIST 是提供信号塔服务的单一机构，这个特性更加优越。

还有更深层次的问题，虽然这个问题看起来可能微不足道，那就是，我们是否可以找到一个方法，它可以使一个现实世界的观测数据——例如温度、太阳黑子图——转变为一个数据字符串，并且它还需要保证每一个观测者都获得相同的字符串？我们可以尝试数字化这个观测数据：比如，我们用华氏度来描述温度，并使用第一个十进制数位作为信号塔的输出，但是除非每一个观测者的温度计都是不可思议的精准，否则就会出现下面的情形，有些观测者读取的温度（比如）是 62.7，而另外一些人读取的则是 62.8。目前看来，不管我们选择哪种自然现象或者采用哪种协议，我们都会遇到这种“极端情况”。对于一个密码学信号塔，即使测量值出现非连续性的可能性非常小，那也是无法接受的，因为这有可能使得 PRG 产生的随机数变得完全不同。

金融数据

还有一个类似的想法是使用金融数据作为数据源，比如股票市场价格指数。同样，这些是公开的可观测的数值，而不像自然现象，这些金融数据本身就是用数字来呈现的，所以不会因为观测者不同，产生数据不一致的问题；同时，我们有很好的理由相信，股票价格的小波动是很难预测的：如果你可以预测纽约证交所某一只股票的交易价格，并且精确到美分级别，你就会成为一个非常赚钱的日内交易员。某些人可以通过买卖股票，去操纵股票价格到某一个特定的数值，但这需要巨大的成本。

但是，这种方法也需要依赖某一个信任方，比如股票交易所。即使股票交

易所本身有很强的意愿去建立自己的诚信，但如果让它们操纵一个非常有利可图的乐透，那么它们就有可能去尝试改变股票价格（例如，通过插入自己的买卖单）。

截至目前讨论的方法，似乎很难去规避信任方的问题，而这个信任方却是可以在整个过程中的某些关键环节对结果施加重要影响的。

用比特币作为一个“信号塔”

幸运的是，把中央权威从数据交换协议中剥离出来，在之前认为几乎是不可能的任务，而比特币是很有希望实现这一任务的技术，而这也是本书的中心思想之一。我们是否可以把比特币作为一个生成随机数的“信号塔”呢？我们想要从比特币的区块链里摘取随机数，与此同时保留比特币所有去中心化的特点，正是这些特点使比特币如此吸引人。

我们回忆一下，矿工必须计算大量的随机哈希函数来找到一个有效区块。或许这意味着没有人可以不经过挖矿工作就能预测或是影响下一个区块的生成。当然，任何一个区块的哈希函数结果的最初几个字节都是零，但是在合适的假设下，唯一可以预测剩余位数的比特值的方法可能是找到一个胜出有效区块，然后选择性丢弃它（见图9.9）。

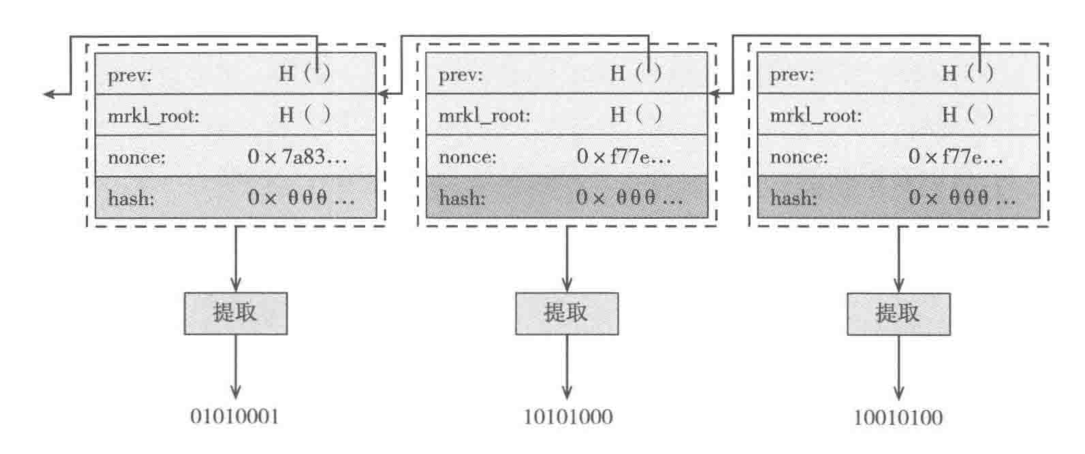


图9.9 比特币像灯塔一般

注：我们可以通过使用随机提取功能，提取公共随机数据，标注区块链上的各个区块。

这样一来，把区块链变成一个随机数“信号塔”成了一件简单的事。在区块链上的每一个区块上，我们在区块头部设置一个“随机数抽取器”。随机数抽取器，其实就是一个哈希函数，这个哈希函数把所有的输入随机熵均匀地压缩成一个随机字符串。每次只要发表一个区块，我们就有了一个新的随机信号输出。

评估比特币“信号塔”的安全性

假设你参加一个乐透抽奖，这个抽奖的结果是由未来将要产生的、一个预先设定的、位于高度 h 的某个区块的输出所决定的。这个乐透抽奖有 N 个参与者，每个参与者都下注了 B 个比特币，如果你也是一名矿工，我们再假设，你幸运地找到了一个区块 h 的函数解谜的答案，你就可以选择发表或是不发表这一区块。如果你不喜欢从这个区块里产生的抽奖的结果，你可以轻易地丢弃这一区块，然后让其他找到这个区块的人来决定这个抽奖结果，但同时，你必须放弃因为找到这个区块所带来的收入。

让我们计算一下，下注的数额 B 值需要多大，才值得考虑放弃区块本身的奖励。如果你成功地找到了一个位于高度 h 的区块来决定抽奖的结果，然后意识到，如果你发表了这个结果，你肯定会输掉抽奖；如果你扔掉这块，你还是有 1/N 的概率来赢得 B×N 比特币，这就意味着，如果你期望的抽奖奖励（1/N）×B×N 比特币比挖矿所获得的 25 个比特币奖励要大的话，那么放弃区块奖励是合理的（在 2015 年，如果不考虑交易费的话，区块奖励大概是 25 个比特币），所以如果 B 大于 25，这种丢弃策略就是有利可图的。在 2015 年中旬，25 个比特币价值大约在 5 000 美元左右。所以如果每个玩家下注不到 5 000 美元的话，并且假设每个玩家都是理智的，那么这个乐透抽奖是可以抵抗放弃有效区块这种攻击的。

另一大优点是这是一个完全去中心化的信号源，没有一个中心化的信托方。比起其他几个信号源的方案，它的处理速度相当快，大概每 10 分钟就产生一个输出。然后通过上述的简单模型，我们可以估计一个攻击者想要操纵信号源输出所需要付出的代价。

用比特币作为信号塔办法的一个缺点是，不能精确定时。比如，假设我们想要在明天正午读取这个信号源的值，但我们无法知道哪个区块在哪个时点会生成最新的一个区块。虽然平均来说，在正午之前或之后的10分钟内一定会有一个区块被公布，但这还是会有误差。如果我们想降低目标区块在一个短分叉事件中丢失的可能性，我们还要对有可能发生的延迟有所准备。在比特币世界里，通常情况下要等6个区块（60分钟）后，才能确信这个信号值是真正地被确认了。

另一个缺陷是，相对来说，操纵这个信号值所需的代价可能还是太低。如果我们用这个方法来实行NBA选秀，由于其中可能涉及几千万美元利益，球队顿时就有了贿赂矿工来操纵选秀过程的动力。所以，当涉及巨额资金时，这个方法是否有效仍值得探讨。

最后，我们的安全评估忽略了一些现实生活中的因素。比如，对于加入某一个矿池的矿工来说，丢弃一个有效区块并不会让他损失很多钱，因为他们是根据贡献算力的比例，而不是区块来领取奖励的。所以，比特币信号塔目前还是一个有趣但没有被证明的想法。

脚本语言对信号塔的支持

如果我们扩展比特币的脚本语言功能，加入一个特殊的**操作码**（opcode）来读取比特币信号呢？按照最初的设计，现在的比特币脚本语言没有任何办法去实现任何随机性，因为矿工必须验证脚本，而且一个脚本的有效性需要获得所有矿工的认可，但如果我们用了信号塔产生的随机数，由于这是一个可被证明的公共随机数，把这个随机数加入交易脚本中，矿工就容易随机性地达成共识。

假设我们有一个操作码可以做一个随机的决定，这个决定是基于上个区块的信号塔输出的。我们可以把整个复杂的抽奖数据协议用一个脚本来替代——读信号塔的随机数值，然后把该输出分派到n个密钥中的一个。这需要有多回合的数据协议安全保证或是有时效的函数约定。

这个想法的一个缺点是，为矿工操纵抽奖提供了可能性，如果他们发觉挖

到的这一个区块里的交易会让他们输掉这个抽奖，他们就会简单地将抽奖交易延迟至后面的一块出现。但是我们可以对信号塔的操作码做一个小小的调整来防御这类攻击，也就是说，你不是用上一个区块，而是使用某一个特定高度的区块所产生的信号塔随机数。

9.5 预测市场和真实世界的数据源

作为本章的最后一个论题，我们现在来看一下如何利用加密数字货币，以去中心化的方式来实现一个预测市场，与此相关，如何把真实世界的数据导入比特币系统。

在预测市场中，人们可以在一起对未来的事件进行下注，比如体育比赛或是选举。对于事件发生的每一个结果，参与者可以买卖和交易相应的"份额"。

表 9.1 2014 年世界杯期间球队选择的预测表（数字代表在每个阶段下注某支球队捧杯所需花费的美元）

球队	德国	阿根廷	巴西	美国	英格兰	荷兰
赛事之前	0.12	0.09	0.22	0.01	0.05	0.03
小组赛后	0.18	0.15	0.31	0.06	0.00	0.05
半决赛前	0.26	0.21	0.45	0.00	0.00	0.08
决赛前	0.64	0.36	0.00	0.00	0.00	0.00
决赛后	1	0	0	0	0	0

注：押注美国队赢得世界杯的价格在美国队小组表现出色后，从 1 美分上升到 6 美分。当巴西打进半决赛后，其赌注价格上升到了 45 美分，而当巴西队输掉半决赛后，这份赌注变得毫无价值。最后只有德国队的赌注才有价值，因为他们赢得了冠军。

我们用一个案例来详细解释一下预测市场背后的概念，使其更加清晰。2014 年的世界杯在巴西举行，假设有一个市场，你可以买卖每个队的赌注。最终冠军队的赌注是 1 元，而其他队的都是 0。比赛开始之后，根据市场认为每个队最后能赢得冠军的概率，每个球队的赌注都会有一个价格。表 9.1 就是五个

队的赌注价格情况。

在比赛前，德国队赌注的交易价格是12美分，意味着市场觉得德国队大约有12%的机会获得最后的冠军。当比赛进行的时候，这些赌注价格会上下波动，反映了市场参与者对每个队最终获胜的信心。

在我们的案例中，英国队赌注本来的交易价是5美分，但后来变成了0，因为英国队没有小组出线，他们已经不可能取得最终的胜利，价格也相应地反映了这一点。与之相反的是，美国队最初被认为很难从小组出现，但是结果他们的小组赛表现却相当不错，如果你在最初美国队赌注价格非常便宜的时候（1美分）买了它，并在它出线并变成6美分的时候马上卖出，你就可以拿回6倍于你最初的投资，而不需要等到全部比赛完了之后再卖出。虽然美国队最后没有赢得世界杯，但你还是可以在美国队小组赛表现抢眼的时候，通过市场对美国队的信心调整来获利。

半决赛的时候只剩下四支队伍了，由于美国队与英国队都被淘汰出局，所以他们的价格都是零。每个剩下的队都有一个高价位，他们的价格之和是1美元。巴西队价格最高，因为那时市场认为巴西最有希望赢。当巴西队输了半决赛的时候，它的价格马上变成了零。在两个小时内，市场对其信心就发生了戏剧性的改变。你可以对巴西队进行一个卖空或者去买其他球队。

到了决赛的时候，只剩下两个队。它们的价格总和还是1美元。当然到最后，德国队获得了最终的胜利，也只有德国队的赌注最终有价值（1美元）。

当然还有一个获利的办法就是，你在最初就以12美分的价格买下德国队的赌注，然后一直持有到最后，直到德国队获胜。这基本上是传统体育博彩的机制——你在比赛前下注，然后在比赛胜利后收钱。但在一个预测市场里，有很多其他办法可以进行博彩和盈利。你可以在任何时间对任何球队下注，你是否可以获利完全取决于你准确地预测市场信心的转变，而与最后结果无关。

这里有另外一个案例，这次是一个完全真实的预测市场案例。在2008年美国大选之前，Lowa电子市场允许人们购买份额下注奥巴马或者麦凯恩获取最后的大选胜利。如图9.10所示，奥巴马的价格显示为实线，麦凯恩的则是虚线。你可以看到，随着竞选活动的开展，人们对谁将最终获胜的信心是波动的，但

是到了大选前的前一天，奥巴马当选的概率达到了90%，在最终投票之前，预测市场对结果的预判基本上已经确定了。

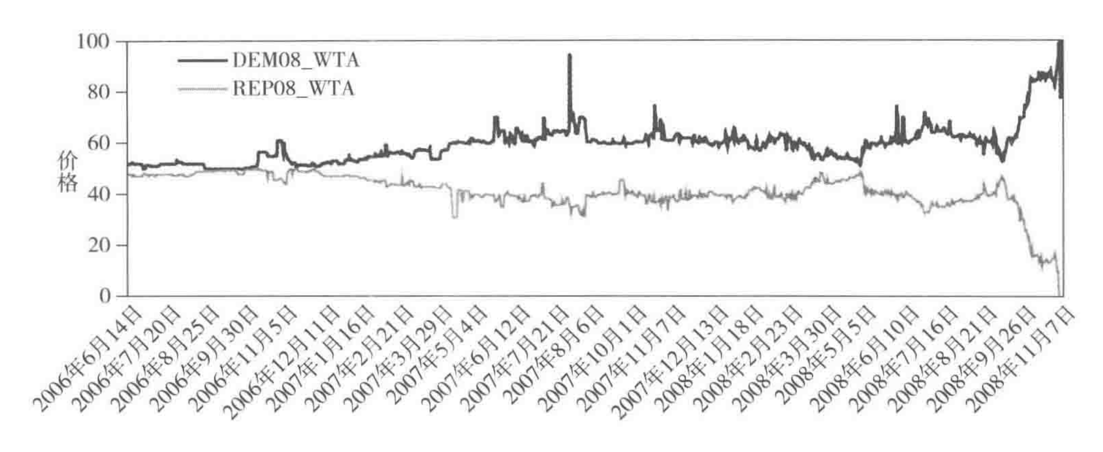


图 9.10　预测市场份额

注：对于2008年美国总统大选预测份额的价格走势图。

资料来源：Lowa 电子市场

预测市场的力量

经济学家对预测市场普遍非常热情，关系到预测未来事件的相关信息通常都是比较分散的，由于预测市场提供了一个平台，每一个参与者都可以利用他们的相关知识获利，因此它形成了一个非常好的可以汇聚这些信息的机制。在恰当的经济模型中，股票的市场价格可以被解读为最终结果发生的概率，虽然真实的预测市场还是会存在偏差。根据经验来看，预测市场比类似于投票调查和专家论坛之类的其他预测方法更加准确。

然而，预测市场还是面临很多合规性方面的不确定性和障碍，在2013年碰到合规性问题并被关闭之前，Intrade是美国最受欢迎的在线预测市场。许多经济学家对此很失望，因为他们觉得我们损失了一个非常有价值的可以揭示未来的社会性工具。

去中心化的预测市场

如何建立起一个去中心化的预测市场？我们必须将几个重要的任务去中心化。我们需要一个方法来搜集和发放资金，以使这个预测市场能够建立；我们还需要一个方法来确保执行正确金额的资金发放；我们特别需要一个去中心化的仲裁机构，仲裁是一种流程，用来判决哪个结果是真正发生的。大多数的体育比赛和国家选举，最后结果都显而易见，谁赢谁输，一目了然，但还是有许多灰色地带。我们还需要把下单登记系统去中心化管理，其实就是让参与方直接找到交易的另一方。接下来，我们将按顺序来讨论这些挑战。

让我们来设计一个假想中的被称为“未来币”（Futurecoin）的另类币，专为预测市场所用。我们需要设计一些交易类型实现专为预测市场而设的功能，如图 9.11 所示。

CreateMarket（event_id，arbitrator_key，num_outcomes）
　　创造一个新的预测，明确仲裁者和参数
BuyPortfolio（event_id）
　　为每个未来币的结果购买一份赌注
TradeShares（...）
　　将赌注转化为未来币
SellPortfolio（event_id）
　　将每个未来币的结果兑现赌注
CloseMarket（event_id，outcome_id）
　　通过将特定情境的押注所得转入一个新完成的未来币，并注销所有其他情形的赌注，来结束特定预测市场
（outcome_id 是一个介于 1 和该特定输出 num_outcomes 数之间的整数）

图 9.11　未来币中的新交易类型

注：未来币是一个实现了去中心化的预测市场的假想的另类币。

CreateMarket 指令允许任何用户制造出一个针对任何事件的预测市场，然后授权一个特定的仲裁者（也就是一个公钥）来宣布这个事件的结果，以及一系

列可能的结果。event_id 是一个任意的字符串，可以把不同的交易关联起来指向同一个市场。未来币既不关心 event_id 特指哪个实际事件，也不关心结果是什么，当然在系统里也没有办法来具体定义。用户必须保留这些来自市场创建者的信息（通常这些创建者也是仲裁者）。我们会谈到这个仲裁机制的不同选项。

支付和清算

利用这个 BuyPortfolio 指令交易，你可以对不同事件的预测组合进行投资下注。以未来币作价，你可以购买每个事件可能发生的预测结果。比如，我们下注 2014 年的世界杯，32 个参赛队都有可能赢。你可以用一个未来币购买 32 个份额，每个队一个份额——因为最终只有一个队会赢，这些份额的总价格就是一个币。任何一个参与者都可以单方面地创建一个 BuyPortfolio 指令而无须一个对手交易。这个交易实际上是利用用户提供的一个未来币的消耗，以此制造出一个新的份额输入，并分派给每一个可能发生的结果。还有一个交易类型叫作 SellPortfolio，你可以卖（或消耗）每一个对应不同的结果的份额，以此赎回一个未来币。未来币和每一个结果对应的份额可以进行互换。

你可以用份额来换未来币，只要可以找到交易对手，你也可以用一种份额去换另一种份额。下面的案例就更加有趣了，你可以用一个未来币购买每一种可能发生的结果的份额，然后把那些你认为不太可能发生结果的份额的卖掉，对于那些你认为没有什么机会获胜的球队，你可以把相对应的份额卖给其他对此有兴趣的人。一旦你做了这些，你的投资组合就不再是每个队均分的，你也就不能再自动赎回一个未来币了，取而代之的是，你必须要等到最后的结果出来之后才能赎回你的份额——如果你所押的球队没有最终获胜，你可能什么都拿不回来。另一方面，你也可以直接从交易中获利。你可以购买一个平衡的投资组合，等待价格变化，然后出售所有的份额以换回更多的未来币，这些未来币可以用来和比特币或者其他货币进行兑换。

预测市场的仲裁

如何用去中心化的方法来实现仲裁呢？如何做出判断并宣布胜者和定价，

然后胜者可以赎回他们所赢得的份额？最简单的系统就是找一个信得过的仲裁员，也就是上面所说的 CreateMarket。任何参与者都可以发起组织一个市场，在这个市场里他就是仲裁员（或是指定某人为仲裁员）。他们可以创建一个交易，然后宣布发起组成了一个市场去预测世界杯的比赛结果，他们会决定谁是最后的获胜者，如果你相信他们，你就可以接受他们在 CloseMarket 交易上的签名作为最后判决的依据。

就像其他的市场一样，我们可以想象，经过一段时间后，有些实体慢慢地建立声誉并成了可信任的仲裁者。然后它们就会主动维护它们有价值的声誉并做出公正的仲裁。但是，一旦潜在的获利大于其声誉价值，就会存在风险，也就是它们有可能会去操纵一个预测来获取巨额收益，这对预测市场而言是非常危险的。举例来说，在世界杯的预测市场里，即使阿根廷队事实上输掉了比赛，但是仲裁者还是有可能宣布阿根廷队获胜。如果仲裁者自己买了大量的阿根廷队获胜的份额，他可能会通过操纵这个结果赢足够多的钱，而不在乎毁掉他的名誉。

我们可以有一个更加去中心化的仲裁系统吗？一个选择是设定多个仲裁者，然后基于多数人的决定做出判决，或者基于投票结果——要么由所有在市场上拥有份额的用户进行投票，或者由加密数字货币的矿工进行投票，这些投票方案通常也会要求对投少数票的人进行相应地惩罚。但这些方法都有很多问题，所以我们也不知道它们在实际运用中是否可行。

现实是复杂的。除了仲裁者可能作假的问题之外，事件结果的判断也可能存在争议。我们最喜欢的一个案例就是 2014 年的超级碗比赛，超级碗上有一个传统，胜利的球队会将一桶佳得乐（Gatorade）饮料倒在他们主教练的头上。人们想要去对获胜球队用来庆祝的佳得乐的颜色进行预测，这种预测市场由来已久。在 2014 年，预测结果包括黄色、橙色和其他佳得乐饮料所有的颜色。但是在那一年，一个前所未有的结果出现了，很难去决定最终的结果是什么。当海鹰队（Seahawks）获胜的时候，球员们把一桶橙色的佳得乐倒在了主教练彼得·卡罗尔（Peter Carroll）的头上，仅仅过了一会儿，另外一些球员又倒了另外一桶黄色的佳得乐。

如果你主持了这么一个预测佳得乐颜色的预测市场，你会怎么处理这个情况？最终结果应该是橙色，还是黄色，还是两个都算？实际情况是，好几个体育博彩服务提供商为了保持自己的声誉，即使他们因此而损失一些金钱，但为了获取客户对他们的信任，还是决定支付奖金给所有预测橙色和黄色的用户。

当然，在一个去中心化的市场里，这种做法并不容易，因为你不可能无中生有地创造出更多的资金去支付两种结果的赢家，很可能是仲裁者让预测橙色和黄色的双方平分奖金，最终这两种份额的价格都会变成0.5而不是1.0。为了避免这种复杂情况，你可以一开始在合约里定义清楚，但是你不可能确保你能考虑到所有的可能性。这个案例让我们深刻地意识到，仲裁是个社会问题，通过技术手段是无法完美地解决这个问题的。

实时数据供给

仲裁这个概念引导出一个更加广义的概念：扩展虚拟货币的功能来宣告现实社会里的事实。我们称之为实时数据供给。一个典型的预测市场的事件的事实，比如谁赢了选举、某只股票或者某个大宗商品的当天价格或现实世界里有价值的数据。只要比特币里有了这些数据，脚本语言就可以将其作为输入。比如，一个脚本可以将现货金属铜的价格加载在堆栈里，然后据此价格做出决策。[①]

只要存在一个值得信任的实时数据供给，我们就可以对体育比赛结果或是期货市场的价格进行预测投资和自动结算。预测市场只是其中一个应用而已。你可以通过对相反的两个结果都进行预测投资，以实现在你的投资组合里加入风险对冲。你还可以派生出一些金融衍生产品，比如目前金融市场上常见的远期合约和期货合约。如果这些都能通过比特币来实现，岂不是更好？

我们可以把如何在比特币（或是其他另类币）里用技术手段来表现现实社会事实这个问题，和我们如何建立对数据供给的正确性的信心这个社会问题分离开来。

一个聪明的把数据供给编码到比特币的方法叫作现实密钥（reality keys）。

---

① 技术细节请参阅第3章相关内容。——译者注

在这个系统里，仲裁者制造出一对密钥，并用该密钥对他们所感兴趣的所有事件的所有结果进行签名。一个密钥代表“是”，另一个代表“否”。他们在注册登记事件的时候先发表公钥，然后当结果确定的时候，再发表那一对密钥里的私钥。如果爱丽丝和鲍勃共同对一个事件进行预测，他们可以把各自的保证金发送到一个比特币输出，爱丽丝可以使用她自己的私钥和“是”这个密钥进行联合签名以提取这个奖金，鲍勃可以使用他自己的私钥和“否”这个密钥进行联合签名提取。这就很好地实现了公正地使用数据供给作为脚本输入的目标，使得上述预测保证金的应用得以实现。值得注意的是，仲裁者不需要知道，也无须参与到爱丽丝和鲍勃之间的特定预测保证金中去。

交易委托

预测市场的最后一个重要环节是一个去中心化的交易委托，这也是一个通用概念，如果能实现，将会使很多的应用设想变为可能。交易委托是什么呢？在一个真实的预测市场里，或者是大多数金融市场里，并没有一个统一的市场价，通常在交易委托中会有买入价（bid）和卖出价（ask）两种，买入价是指愿意购买份额的参与者所出的最高价，卖出价则是愿意出售份额的参与者所出的最低价。通常卖出价会大于买入价（否则市场就会对此进行撮合，至少其中的一个交易委托将不会出现在列表中）。一个想要购买份额的参与者可以立刻以卖出价购买，而一个想要出售的参与者则可以立刻以买入价出售，这个交易被称为“市价委托”，对应于“限价委托”——交易委托被设定为一个特定的价格挂在交易委托列表中，这些交易委托将会按照限定的价格（或者高于限定的价格）执行。

通常这是由一个中心化的交易委托服务提供商（通常是一个交易所）来实现的。但问题是，就像许多中心化的服务所面临的问题一样，如果这个交易所不诚实的话，它可以通过损害用户的利益来获利。比如，一个交易所收到了一个买单，它们自己可以先在最好的卖出价的时候下单买入，然后马上再在高位卖出，赚取中间的差额。这也叫作预先交易（frontrunning），指的是交易商利用得知客户买卖证券动向的机会，抢在客户发出买卖指令之前为牟取利益而进行

交易的违规行为（例如在股票大量交易前，在期权或期货市场进行相应交易），这是一种金融犯罪行为。中心化的交易委托需要执法部门来监管，来防止这种预先交易的行为，以确保系统诚信的公信力。

在一个去中心化的交易委托里，我们不能依赖强有力的执法部门。但还是有一个较好的解决方案：我们不再将预先交易称为犯罪，然后再想办法去防范，我们称之为一个特性。这个想法是，任何人都可以通过广播交易的办法把限价委托提交给矿工，只要买入价比卖出价高或者相同，矿工就能够撮合两个交易。这个矿工只需把两者之间的差额留下作为交易费即可。这样一来，矿工就没有动机去做所谓的预先交易，因为与此相比，预先交易不可能赚得更多。

这是一个很简练地建立去中心化的委托交易的办法。其最大的缺点是交易者必须支付给矿工费用。为了避免支付这种交易费，交易者们可能会提交偏向保守的交易委托，不会在开始时就透露他们愿意成交的最高或者最低价位，这会使得市场变得不是很有效率。我们现在还不知道，这种让矿工撮合交易的交易委托方法在现实操作中是否可行，但看上去这是个不错的主意。

总结一下，现在比特币可以作为很多种应用的平台，但对于某些应用，比特币也没有更好的发展了，比如，对于实现一个安全的去中心化的预测市场，或者是一个去中心化的交易委托系统，比特币并未提供所要求的全部特性。但假如我们从头开始，忘掉硬分叉或是软分叉，忘掉对比特币增加新功能所遇到的挑战，那又当如何呢？自2008年比特币面世以来，我们对比特币有了越来越多的理解和认识，为什么我们不可以设计一个全新的更好的数字货币呢？

我们将在下一章讨论已经尝试这么做的另类币概念，我们将会探讨所有有前途的想法以及开发一个全新的加密数字货币所面临的挑战。

延伸阅读

我们看过的两种文件的项目材料和说明书，可以参与交易对手方条款说明。您可以通过如下网址阅读：

https：//github. com/CounterpartyXCP/Documentation/blob/master/Developers/

protocol_ specification. md.

OpenAssets Protocol 可通过如下网址阅读：

https：//github. com/OpenAssets/open-assets-protocol.

我们描述过的安全多方抽奖协定（The secure multiparty lottery protocol）可以参阅如下论文：

Andrychowicz, Marcin, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. “Secure Multiparty Computations on Bitcoin.” Presented at the 2014 IEEE Symposium on Security and Privacy, San Jose, CA, 2014.

您可以通过如下网址阅读：

https：//eprint. iacr. org/2013/784. pdf.

经济学家们预测市场的能力的研究，请参阅如下论文：

Wolfers, Justin, and Eric Zitzewitz. “Prediction Markets.” Paper w10504. Cambridge, MA：National Bureau of Economic Research, 2004.

Arrow, Kenneth J., Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, et al. “The Promise of Prediction Markets.” *Science* 320, 2008.

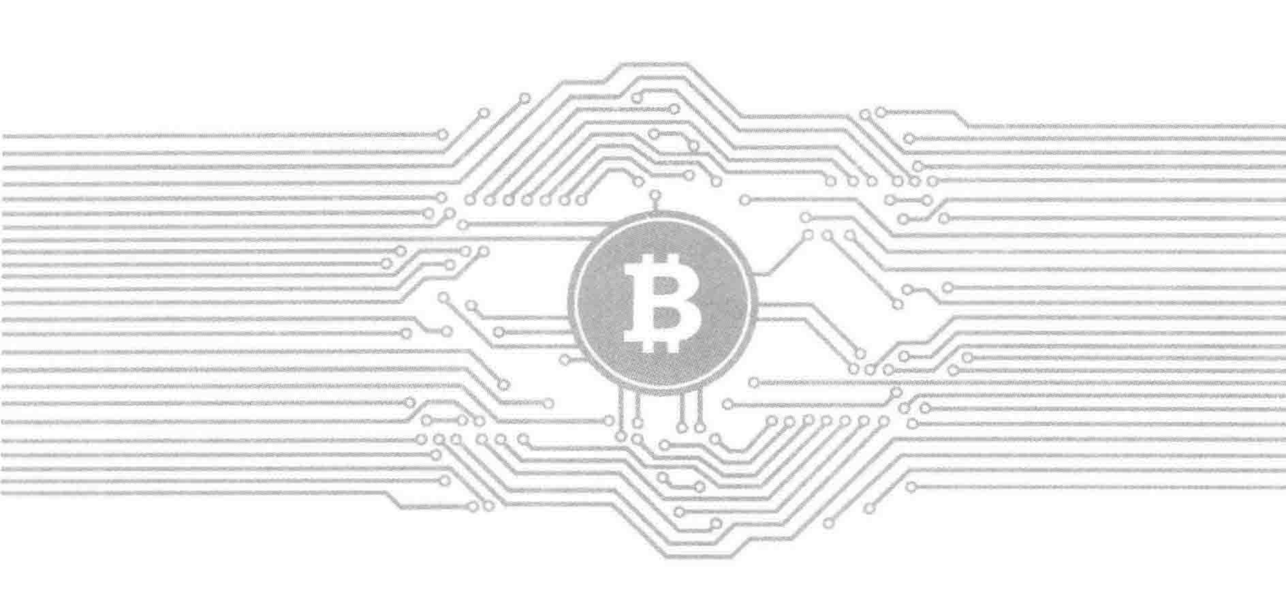
我们讨论过的预测市场设计的相关内容，可以参阅如下论文（由多位作者合著）：

Clark, Jeremy, Joseph Bonneau, Edward W. Felten, Joshua A. Kroll, Andrew Miller, and Arvind Narayanan. “On Decentralizing Prediction Markets and Order Books.” Presented at the Workshop on the Economics of Information Security, State College, PA, 2014.

您可以通过如下网址阅读：

http：//www. jbonneau. com/doc/CBEKMN14-WEIS-decentralizing_ prediction_ markets. pdf.





第10章

另类币和加密货币生态系统

BITCOIN
AND
CRYPTOCURRENCY
TECHNOLOGIES
A Comprehensive Introduction



比特币，尽管是非常重要的组成部分，但它只是范围更广泛的数字生态系统中的一种，该生态系统的其他货币也与比特币相似，我们称之为**另类币**。本章中，我们将探讨另类币及**加密货币生态系统**（cryptocurrency ecosystem）。

10.1 另类币的历史和诱因

2009年1月，比特币诞生。2011年年中，还未到两年，第一个基于比特币的衍生货币——**域名币**就出现了。2013年，另类币出现爆炸式增长，迄今为止已有数百个（见图10.1）。由于没有明确的统计标准，我们无法给出确切的数字。举个例子，如果有人宣布创造了一种另类币，可能也公开了源代码，但无人挖矿也无人使用，这种货币是否需要纳入统计范围？此外，有些另类币，在其诞生初期是有人使用的，但后来很快就无人问津了，这类货币是否也需要纳入统计范围？

而且，我们也不清楚如何区分另类币和传统数字加密货币。早在比特币出现之前，就有多种数字加密货币的方案和系统，这些货币并不能称作另类币。许多另类币借用了比特币的概念，它们通常是直接复制其基础代码或是使用部分代码。有些只对比特币做了极小的改动，例如只改变一些系统参数值，保留比特币开发者后续所做的所有变更。截至目前，所有已知的另类币都是从一个新的创世区块开始，都有自己独特的交易历史，而不是从比特币历史交易记录

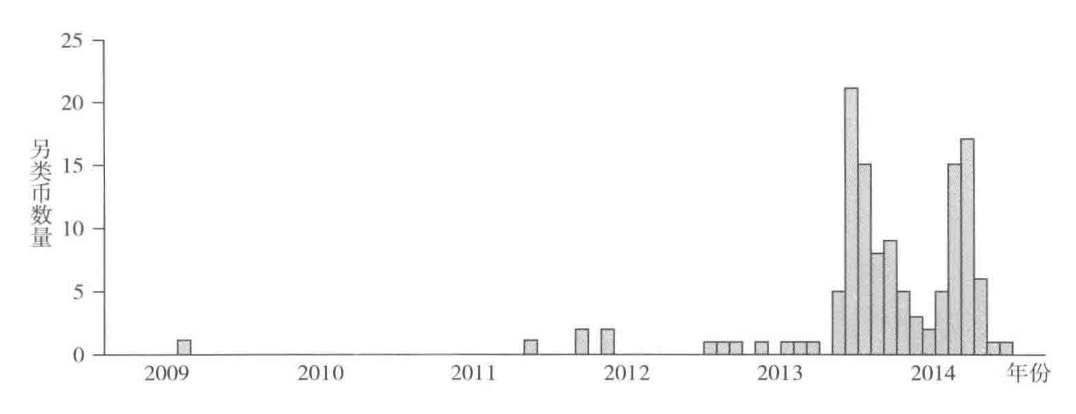


图 10.1　每月创造的另类币量

注：仅指通过创世区块创建的另类币。

中的某个区块进行分叉，进而演化出自己体系的。为了研究和学习，我们并不需要另类币的精确的定义，而是把将所有在比特币之后诞生的加密货币笼统地称为另类币。

我们将简要提一下非另类币系统，如瑞波公司（Ripple）和恒星公司（Stellar），它们属于第2章中所介绍的传统分布式共识协议。这类系统为了达成共识，模型中每个节点都有自己的标记并且需要知道其他节点的标记。当然，比特币建立共识的模型与此截然不同。在瑞波公司和恒星公司中，共识协议支持支付清算网络并且在每个体系中有其自己的货币。尽管这些特性和另类币类似，但在本书中并未将它们作为另类币考虑。

发行另类币的原因

每种另类币都有自己的故事。一种另类币之所以存在，就是因为它有别于其他另类币的特点。最简单的情况下，一种另类币只是修改比特币内置的参数，比如修改区块的平均时间间隔、区块大小限制、创造回报的计划和货币的通货膨胀率等。

当然也有更复杂的技术上的差异，这种情况更有趣。例如，可以对脚本语言进行扩充以增加交易种类和安全属性，可以采用与比特币完全不同的挖矿方式以及**共识算法**（consensus algorithm）。

有时，为了支持一个主题或者社区，通常是需要给社区中的成员赋予一个特定的角色或权限，就会有一种比特币被创造出来。本章将在最后的部分研究此类相关案例。

如何创造一个另类币

我们首先考虑一下另类币在创建过程中及创建后所涉及的内容。正如前文提到的，创造一个另类币就是建立一个全新的参照体系，最常见就是通过复制修改现存的成熟另类币或者比特币本身。最容易的部分就是加一些技术特色或者修改一些参数使之更好用。曾经有一个网站 coingen. io，只收取一些费用，就会自动产生一个另类币。你只需要自己设定各种参数，比如区块产生的平均时间、需要的工作量证明算法、另类币的名字以及 3 个字母的货币代码和标志。完成设置后，只要轻轻点击鼠标，就能下载一份根据你的需要修改的比特币源代码，你就拥有了自己的另类币，接下来你就可以和别人马上开始运用这个另类币了。另类币最困难的部分在于如何让别人逐步接受并使用你的另类币。通过复制并修改源代码，你可以对外发布新的另类币，在刚诞生时，没有人会使用这个货币，由于没有人想拥有这个货币，因此它毫无价值，又由于没有挖矿的人，它也不安全。本书第 7 章，我们介绍过比特币系统的利益相关者：开发者、矿工、投资者、商家、客户和支付服务商。最终，为了让你的另类币形成规模，你需要吸引这些参与人加入这个货币的生态圈中。

另类币的这些相关群体都是非常重要的，而且它们相互关联，这与创建并推广一个平台非常类似。比如，创建一个智能手机操作系统，就需要用户、设备制造商、手机软件开发者和其他重要的利益相关者共同参与，同时每个角色都需要群体中的其他人的参与。

在另类币中，吸引矿工对另类币来说特别重要，因为如果没有足够的哈希算力做支持，双重支付和复制修改代码就很可能会发生，另类币的安全性就无从谈起。事实上，这种货币可能会彻底崩溃，本章 10.4 节中将会讨论“另类币夭折”（altcoin infanticide）。没有一个简单的方法可以吸引大家接受并逐步推广使用另类币，但是通常来说，当矿工感觉到货币回报值得他们付出时，他们就

会加入。为了吸引矿工，很多另类币都给早期矿工比较丰厚的回报。比特币显然是最早采用这种策略的，后来很多另类币采用了更加激进的激励措施来吸引早期矿工。

而最困难的工作，是让一个社区的人相信这个另类币有价值。正如我们在第 7 章讨论过的，即便对比特币来说，我们也不是特别清楚这个过程是如何自举的。这依赖仙子效应，从而实现自我增强，让人们相信它有价值的过程是如何实现的。这就回到我们一开始提到的，另类币需要有一个好的故事，才能让人相信这个新的另类币将来会有价值，或者是相信其他人会认为这个有价值。如果一个社区对获取另类币感兴趣，矿工就会参与进来。[①] 只要价值被认可，其他重要的元素就会显现，比如在交易所交易以及从开发区块链的工具，到游说团体开发的各种辅助设施和服务。

拉高出货骗术（pump-and-dump scams）

当一种另类币的创始人成功地促成一个活跃的货币社区和一个真正在运作的交易市场时，他们就会变得非常富有。几乎可以肯定的是，他们拥有很多这类货币。这种货币可能来源于，在系统运行最初，哈希算力还不是很高的时候所挖的货币，或者是类似接下来要讨论的，在还未挖矿之前获得的预先分配的货币。一旦另类币的交换价值提高了，创始人就可以选择卖掉他们的货币。

一夜致富的可能性，极大地吸引了有雄心的创业者和风险投资基金，毫不意外地也吸引了骗子。事实上，我们很难区分骗子和创业者。骗子可能会使用各种方法，来夸大一种另类币的潜在和未来收益。他们可能会炒作它的技术优点，伪造底层支持的假象，在市场上推高另类币价格等。

事实上，甚至连非创始人都可以设计这样的骗局。他们可以先买入大量还未出名的另类币，然后说服大众相信该货币还有未实现的增值潜力（也就是“拉高”）。如果成功地拉高了货币的价格，他们就可以通过卖出获利（也就是“抛售”）。此时，很多理智的投资者可能会意识到这是个骗局，然后币值出现断

① 尽管当币种升值快于挖矿的速度时，可能会有风险。——译者注

崖式下跌，导致很多当初购买的人最终血本无归，只剩下毫无价值的货币。这种拉高出货的骗局，在操纵不知名的低价股票的主流金融业务中很常见，在另类币的发展早期也很普遍，那时候用户热情高涨，投资者也无法分辨到底哪一种另类币是真正具有创新性的，哪一种是依靠噱头和推广，但实际是毫无真正价值的。这也导致，截至目前，用户和投资者都厌烦了另类币。

初始分配

在比特币体系中，货币只能以挖矿的方式分配给用户。但是在其他另类币体系里，出于各种考量，除挖矿以外，开发者们还使用其他方法对货币进行初始分配。

开发者可以预先分配货币，也就是说，先预留一部分货币给自己或者其他特定团体（比如预留给开发该货币的非营利性组织）。用这个额外的收获，去激励开发者花费时间精力去创造和激活一个新的加密货币。有时候，也可能会采取更激进的激励方式，即可以对货币进行预售，也就是把这些货币预先卖给其他投机者，换取比特币或者现实中的货币。这有点像投资初创企业：如果投资的另类币成长起来，投机者就会获得大量财富。

寻求各种预分配方法的另外一种动机是，确保早期的货币拥有者来自多个社区，并且他们与货币成功的利益相关。如果矿工太为集中，就会造成资产持有过度集中，这不利于货币的发展壮大。一种比较聪明的分散所有权的做法，就是把另类币发给现有的比特币用户。

在技术上如何做到这一点，即让比特币用户可以自动地分配，并拥有另类币？一种办法是通过第3章谈到过的“销毁证明”：用户只要证明他们销毁了一定比例的比特币，就能要回一个单位新的另类币。用户需要在销毁的时候提供数据证明，比如特殊的字符串来识别某个另类币，这样就可以说明他们销毁比特币的目的，就是为了获取这个新的另类币（见图10.2）。

通过“销毁证明”来分配另类币，也叫作“单向挂钩”或者“价格上限”。另类币可以一对一地配对比特币，并不意味着两者价值相同。这样的配对，确保另类币最多值1个比特币。因为，1个比特币可以换1个另类币，但是反过来不行。

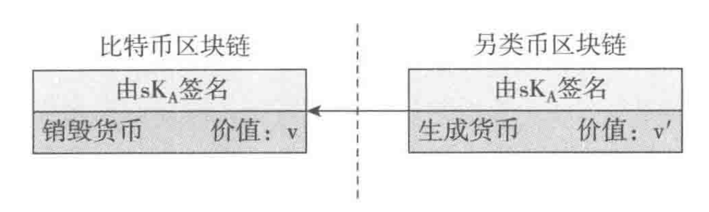


图 10.2 通过“销毁证明”分配另类币

注：另类币提供一个以比特币操作为输入的生成货币（GenCoin）的指令。生成货币的签名用到的私钥，和签销毁证明的私钥是一样的（签名的机制也一样）。这样就能保证，销毁比特币的同一个用户，同时创造了新生成货币。如果汇兑比例是1∶1，那么另类币的价值v′不大于比特币的价值v。

也可以有一些相对没那么复杂的做法：要求用户提供拥有比特币的证明，但无须销毁比特币，也以获得新币。具体来说，另类币体系会指定一个比特币区块高度（也许刚好就是另类币诞生时的长度）。在这个高度的区块里，任何人拥有还没花掉的比特币，就可以按比例得到同样数量的另类币（见图10.3）。通过这种方式，比特币和另类币的价格就无须固定，毕竟比特币并没有通过销毁证明来“转换”成为另类币。

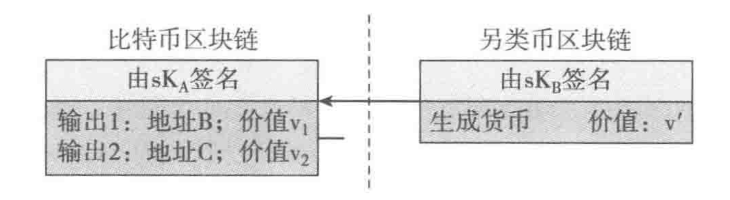


图 10.3 通过证明比特币的所有权来分配另类币

注：生成货币的输入，是特定区块高度下一个或者多个没用过的比特币交易输出。就像正常的比特币操作一样，这些都是通过控制未使用比特币的私钥来进行签名认证的。图中的比特币交易有两个未用过的交易输出，分别为特定高度区块链中的B和C地址。B地址的用户换了另类币，但C地址用户还没这么做。假设汇兑比例是1∶1，新另类币的价值为v′，那么v′一定不能大于B的比特币价值v_1。

当然，为了实现这种分配，另类币的矿工也必须时刻了解比特币的区块链。另类币必须明确什么才算认定的比特币交易。一种选择是要求固定的确认次数，比如6次。另外的一种选择是，在每个另类币的区块中加入最新比特币区块。

这样，比特币的交易立刻可以在另类币体系里使用，而不需要等待确认。这就类似在比特币体系中，交易的输出可以在自身或者下一个区块中使用。我们将在下一节讨论**共同挖矿**（merge mining），一种把比特币和另类币区块链接起来的方式。

最后一种方式是，把已经分配好的货币捐赠出去，这也是扩大货币用户多样性的一种办法。一种方式是支付小费：很多服务允许赠送小费给电子邮箱或者社交媒体账户，这多多少少可以促进接收者了解并参与到这个货币体系中来。接受方收到信息，得知他的托管账户存有小费，然后通过认证邮件地址或者社交媒体账号可以取得这些小费。当然，为获得这些小费，他们还需要安装钱包软件，或者采用其他方式。另外一种可以称作“水龙头”捐赠方式，即任何访问特定网站并输入邮件地址的人，都可以获得一小部分另类币。

10.2　几种另类币的详细介绍

接下来，我们重点介绍几个最早的另类币。

域名币

本书已经介绍过比特币的区块链是一种安全的全球数据库，其对写入的数据具有防篡改保护，并且是永久的。那么是否可以修改比特币的设计，来支持其他安全的全球数据应用（比如域名系统）呢?

为了使这个数据库在非货币方面的应用更加有效，我们首先需要明确几个基本原则。第一，把录入的数据视为域名或数值对（name/value pairs），域名是全球唯一的。这就可以使任何人去寻找可映射到域名的数值，就像哈希链表或者有主索引的数据库一样。为了确保域名的全球唯一性，如果域名和数值对与以前录入的相同，则将其视为对旧数据的更新而非新的数据。

第二，只有首次录入某个域名的使用者，才有权限更新这个域名。这很容

易实现，比如可以把每个域名与比特币地址联系起来，且规定必须用这个地址的私钥，才可以对更新交易签名。

比特币可以实现上述功能，正如第9章中所述，可以把比特币作为只有增添功能的日志来构建叠加货币。由于可以直接把这些协议写入规则中，其他另类币更易于实现上述功能。而且，一旦矿工执行了这些规则，这些规则就是不可改动的，而且不需要每个使用者（如全部节点）自己检查并判断在受到侵犯时该如何处理。它甚至可以实现类似SPV形式的验证：一个轻量级的客户端可向运行全节点功能的服务器提交一条查询（如查域名），服务器则会返回这个域名项的数值以及相关证据，用以证明返回的数值是数据库中最新的数据。

上文简要介绍了域名币。这是一个全球的域名/数值商店，在这个体系中，每一个用户都可以注册一个或者多个域名（需要一定的费用），并且可以更新他们拥有的域名的数值。用户同时也可以把拥有的域名转给别人。事实上，由于域名转让与货币转让交易不可分割，你可以在把域名转给别人的时候，获得几个单位的域名币。通过这种方式把域名卖给从未谋面并且未建立信任关系的人是安全的。虽然截至2015年，域名币还无法支持安全简单的客户端，但是支持这个扩张功能的提议已经被提出来了。

域名币的目的是提供一个去中心化的域名系统DNS，在DNS数据库里，名字即域名，数值对应IP地址。目前还无法在普通浏览器里默认使用域名币，但是通过下载相关插件，这个插件就会在域名币的注册系统而非传统的DNS中查找IP地址，这样，用户就可以在像火狐（Firefox）或者酷容（Chrome）此类的浏览器中访问如example. bit这样以 . bit结尾的域名了。

域名币不仅在技术上而且在历史地位上都是很值得一提的。它诞生于2011年4月，仅比比特币晚两年，是第一个被创造出的另类币。其特点是共同挖矿，本章10. 4节将就此做进一步讨论。

截至2015年，域名币并未被广泛使用。大部分的注册域名都被一些投机者抢先注册，他们希望通过卖域名获利（现实远未如此）。支持域名币系统的人认为，不应该让现有DNS体系将互联网核心组成部分的控制权，过多地交

由单一机构来管理。可以想象，这种观点在比特币社区中也很流行。但是主流用户对于用其他方式来替代 DNS 并不热衷，因此，域名币这种杀手级应用无法普及。

莱特币

莱特币（Litecoin）诞生于2011年，在域名币之后。在过去的几年里，无论是从综合流行程度或是用户基础看，莱特币都是另类币中的领头羊。它也是被模仿修改最多的货币。事实上，莱特币被模仿修改的次数超过了比特币。

莱特币和比特币在技术上的主要区别是：莱特币用的是第8章讨论过的基于 Scrypt 算法的刚性内存解谜（memory-hard puzzles）。当莱特币出现的时候，比特币的挖矿还在 GPU 时代，所以当时莱特币使用刚性内存解谜，目的是替代 GPU。一开始发行时，还可以用 CPU 在莱特币中挖矿，虽然那时候比特币早已无法使用 CPU 来挖矿。但是后来，莱特币也无法阻止挖矿的层层升级，从 CPU 到 GPU 再到 ASIC。每次莱特币挖矿的升级，都比比特币花费的时间更长。其中原因，也许是因为莱特币的谜题，用硬件去解更难，或者由于莱特币币值交换比例较低，使得矿工缺乏动力。

不管是何种原因，从 CPU 升级到 ASIC，就挖矿功效的改进效果来看，莱特币与比特币类似。从这点来看，莱特币并没有达到原先设计的目标：通过维护 CPU 矿工社区，创造出一个用 CPU 挖矿的分布式体系。但是，重要的是，这个理念虽然失败了，它依然吸引并保持了众多的追随者。如今，莱特币已经改变了其说法，声称由于其并非采纳 ASIC，因此其初始分配更加公平。

莱特币也做了一些小的参数变更，比如莱特币的区块增长会比比特币快4倍，也就是每2.5分钟产生一个区块。其他方面，莱特币都尽可能借鉴比特币。甚至莱特币的更新都跟随比特币，比特币一有任何补丁或者更新，莱特币会同时采用。

狗币

狗币（Dogecoin）也许是迄今为止故事最精彩的另类币。它诞生于2013年年末，其突出的特点不是技术（它是莱特币的翻版），而是社区价值体系：小费、慷慨和非严格的加密货币。它的名字来源于神烦狗（Doge），一只有趣的在互联网流行的日本柴犬（见图10.4）。狗币团队发起过好几个有趣而且成功的广告宣传活动，比如赞助美国纳斯卡车赛（NASCAR）车手，让狗币的图案遍布全车。他们还集资了3万美元，资助牙买加国家雪橇队参加2014年冬季奥运会。有趣的是，这和90年代的电影酷跑（Cool Running）的故事情节如出一辙。



图10.4　狗币的其中一个标志

注：卖点是其有趣幽默，而不是其技术创新。狗币标志，版权为2013~2014年狗币开发者。

由于狗币社区的慷慨大方、宣传活动的推广，加上神烦狗形象在互联网的流行，狗币在2014年一度大受欢迎。很多用狗币的人，之前都不知道什么是加密货币，他们也不需要知道狗币比别的货币好在哪里，就可以主动参与并推动狗币的发展。狗币的成功，说明一个货币的流行也可以通过非技术的方式来实现。遗憾的是，就像很多互联网热点一样，狗币的风靡程度目前已逐渐减弱，其汇兑比率也随之大幅下降。

10.3 比特币和另类币的关系

我们可以用一系列的参照标准，来比较各类不同的另类币的相对规模和影响。

另类币比较

资本市值

传统上来说，资本市值是评估一个公众公司的简单方法，把公司股票价格乘以总股份即可得出资本市值。在另类币领域里，计算资本市值的方法类似，即用每单位货币的价格（通过用最通用的第三方交易平台取得价格）乘以流通中的总货币数。按照这个标准，截至2015年，比特币的资本市值最大，其占了所有加密货币总和90%多的市值。其他币种的排名可能会经常变化，但是大部分另类币的市值都非常小。

不能过分看重资本市值。首先，资本市值并不等于购买所有流通中货币的总费用。这个总费用可能比资本市值高或者低，因为大量的购买会抬高该另类币的价格。其次，虽然资本市值只考虑流通中的总货币量，但是市场参与者应该会考虑未来新发行货币量对货币价格的影响，这使资本市值的估计更加复杂。最后，甚至真实流通的货币量也是无法准确估计的，因为有些货币的主人也许丢失了他们的私钥，但我们并不知道。

挖矿能力

如果两个另类币用同样的挖矿谜题，那么可以直接对比每个货币矿工的总挖矿能力。基于哈希谜题的影响，挖矿能力这个指标经常被称为**哈希速度**（hash rate）。比如，**泽塔币**（Zetacoin）用和比特币一样的SHA－256挖矿谜题，在2015年12月，它的网络哈希速度，是5兆哈希每秒（即5×10^{12}哈希/秒）。

这个数字大约是比特币的十万分之一。如果两种货币用不同的挖矿谜题，比较挖矿能力就很困难，因为谜题需要花不同的时间去计算。而且，专门为某种货币定制的挖矿硬件不一定适用于其他货币的挖矿（包括攻击）。

即便对使用完全相同的挖矿谜题的另类币，我们也可从其随时间而变化的挖矿能力上，得知一些有用的信息。如果是算力的增加，则意味着更多的人加入或者现有参与者升级了更强大的硬件设备；反之，如果是算力的减少，则意味着一些矿工已经放弃这个币种，这通常是一种负面的迹象。

其他指标

有几个其他指标可以用于比较另类币。比如，另类币的汇率变化可以说明其健康程度，当然也可能与其哈希速度变化有关；在第三方交易平台的交易额可以用来测度这个货币的活跃度和公众对它的兴趣程度。然而，有些指标并不一定有用。比如，另类币区块链的交易量就并不能说明什么，因为这有可能是用户在他们自己的账户内通过倒转货币产生的，这些内部倒转甚至可能是自动进行的。最后，我们可以看看有多少商家和支付渠道支持这种货币，因为只有优秀的货币才更可能得到支付渠道的支持。

比特币与另类币互动的经济学视角

比特币和另类币的关系很复杂。一方面，作为加密货币，因为它们都可以用于网络支付，它们相互竞争。如果两种货币提供的功能相近，采用类似的标准、协议和规范格式，那么最终有一方会占优，这就是经济学家所说的“网络效应”。

举个例子来说，在2000年的中后期蓝光（Blu-ray）和HD DVD展开激烈的竞争，争夺DVD标准的制定者。因为受欢迎的游戏机PS3（PlayStation 3）的主机可以当成蓝光播放器使用，渐渐地，蓝光开始变得更加流行。很多电影制作商也由此更喜欢采用蓝光格式，这也反过来推动了蓝光进一步的普及。随着更多的电影采用蓝光格式发行，更多的用户购买蓝光播放器，进一步导致更多的电影采用蓝光格式。同样，如果你的朋友都用蓝光播放器，你也会买蓝光而不

是 HD DVD 播放器，因为这样，你和朋友互换电影会更容易。因此，仅仅用了两年的时间，HD DVD 就成了历史。

谁最终获胜？

早在 HD DVD 被淘汰之前，就有无数相似的技术标准被竞争者迅速取代，从而暗淡地退出历史舞台，如 Betamax 模拟磁带以及俄罗斯标准铁路轨道等。因为网络效应，你可能没听过这些被替代的东西。有时候，胜出者是因为其压倒性的技术优势，就如同尼古拉·特斯拉（Nikola Tesla）的交流电网在与托马斯·爱迪生的直流电网中获得的优势一样。然而，大多数时候，失败一方事实上在技术上更胜一筹，比如 Betamax 磁带就输给了 VHS 磁带，可见网络效应如此之大，以至于微小的技术劣势可以忽略。

以上的推理说明，即使后面跟随的其他货币在技术上更先进，最终只有一个加密货币会占主导地位（目前看来可能是比特币，因为它是当前最流行的货币）。但如果仅仅只是这样看待货币之间的竞争，那就会过于肤浅，加密货币之间的竞争，之所以不像光盘格式之间的竞争那么你死我活，至少有以下两个原因：

首先，用户从一种加密货币转向另一种相对容易，服务商也容易接受多种加密货币，这意味着多种加密货币更易于共存和发展。加密货币的这种特性在经济学上被称为较低的转换成本。相反，DVD 播放器的转换成本很高，因为绝大多数人不需要两个笨重的播放器摆在家里，如果更换到另外一种播放器，也不容易把已经有的光盘转化为另外一种格式。当然，加密货币之间的转换成本也不是完全没有。比如，用户也许已经购买了一个硬件版的钱包，如果转换其他货币，硬件钱包可能无法升级转化。但是正常来说，转换加密货币并同时使用多种加密货币是很简单的。

其次，正如前面提到过的，很多另类币之所以存在，是有其独特的功能基础的。这些另类币并不只是比特币的替代品。它们和比特币功能有交叉，甚至

互补。从这个角度看，可以功能互补的另类币实际上扩大了比特币的用途，而并不仅仅是和比特币竞争。例如，假设域名币成功了，那么比特币的用户在使用比特币的时候，就多了一个选择。

当然，如果把它们之间的关系都理解成愉快的合作共赢，也过于肤浅。一些另类币，比如莱特币，就是想要用更加高效的方式来达到比特币的功能。莱特币的创新功能，其实都可以在比特币体系里实现，或者用相对笨拙的办法来实现（第11章将做进一步的讨论）。支持在比特币体系基础上持续改进的人认为，多种另类币分散了可用哈希算力，从而使每个独立货币不太安全。

相反，支持另类币的人则认为，另类币可以让市场决定什么功能值得拥有、什么系统更加优越等。他们同时还认为，多种另类币系统同时存在的话，可以把任何一个货币系统灾难性的损失控制在一定范围内。他们也指出，比特币开发者高度风险厌恶，不管是通过软分支还是硬分支，加入任何新功能，都是非常缓慢和困难的。相反，在另类币上很容易去尝试新想法。因此，可以把另类币视为比特币新功能的实验田。

实际上，最终结果就是，比特币和另类币的支持者之间既相互竞争也相互合作。

10.4 另类币的夭折与共同挖矿

本节和下节将继续讨论比特币和另类币的技术相关性，而暂时搁置文化、政治和经济因素。

另类币的夭折

截至2015年，比特币的哈希算力让所有任何其他另类币相形见绌。事实上，存在几个势力强大的矿工或者矿池，他们控制的挖矿能力高于所有其他另类币的挖矿能力总和。这样的矿工或者矿池，可以轻松攻击一个小的另类币

(如果他们也用和比特币一样的 SHA-256 挖矿谜题)，通过制造赝品和大规模混乱，最终毁了该另类币。我们称这种现象为另类币的夭折。

用宝贵的挖矿算力去攻击其他货币，并且得不到明显的金钱回报，为什么会有人这么做？以 2012 年**盘旋币**（CoiledCoin）被攻击为例：比特币矿池 Eligius 的总管认为，盘旋币是个骗局，会对整个加密货币的生态系统产生冲击。所以，Eligus 将其挖矿资源全部用在盘旋币上，制造出的区块链把盘旋币几天的交易给对冲掉，同时挖了一条很长的空区块链。这造成了其他盘旋币用户无法再使用盘旋币的服务，也就无法再产生任何新的交易。在盘旋币经历了短暂的攻击后，用户放弃了盘旋币，它从此销声匿迹。在这个案例，以及其他类似的另类币夭折的案例里，攻击者都是出于金钱以外的动机而发动攻击的。

共同挖矿

如果一个另类币复制了比特币的源代码但是没有做任何修改，按道理在这个另类币上的挖矿是有排他性的。也就是说，你可以去试图找挖矿谜题的答案从而找到一个有效的区块链，但是只能给另类币或者比特币，不能一石二鸟。你可以把你的挖矿资源在比特币和另类币上做分配，你甚至可以在多种另类币上分配资源而且随时调整配置，但是你无法让挖矿资源同时服务于多种货币。

在这种具有排他性挖矿的条件下，网络效应会使很多另类币无法实现自我增强式的循环发展。如果你开发了一个新的另类币并成功说服当前的比特币矿工加入你的另类币体系，为此，他们必须停止比特币的挖矿，也就意味着他们会立刻产生相关损失。因此，他们没有动力加入你的另类币体系，也就意味着你的另类币很可能只有很低的哈希算力，也就很容易被其他比特币矿工攻击并夭折。

是否可以设计出这样一种另类币，它可以允许同时在该币和比特币上进行挖矿？为了达到这个目的，则必须创造出包含比特币和该另类币相互交易的区块链，以使这些交易在两个区块链均有效。设计可使比特币的交易出现在其区

块里的另类币，这个并不难，我们可以设计任何想要的另类币的规则。但反过来却很难。如何把另类币的交易放入比特币区块链上？第 3 章和第 8 章已经介绍了如何把任意数据放在比特币的区块里，但是这样做会遇到比特币特有的带宽限制，即其数据传输量非常有限。

然而还是有巧妙的办法：虽然不能把另类币的交易内容放进比特币的区块里，但是可以把另类币的交易概要以哈希指针的形式放入比特币区块中。找一个可以在每一个比特币区块里放入一个哈希指针的办法很容易。具体来说，回想一下本书曾经提过每个比特币区块都有一个特殊的交易，称为**币基交易**，也就是矿工创建新的区块所得的比特币奖励。这种交易的**输入脚本**（scriptSig）区域没有任何内容，因此可以用来存储任意数据（当然也不需要对币基交易进行签名认证，因为没有任何前序交易）。所以在一个共同挖矿的另类币体系里，挖矿的任务就是去计算一类特殊的比特币区块，币基交易的输入脚本区域存有指向另类币区块的哈希指针。

这个区块现在可以身兼二职：对比特币客户端来说，其与任何其他比特币区块没有区别，除了在币基交易中多了一个可以被比特币忽略的哈希值。另类币的用户知道如何解读这个区块：忽略比特币的交易，只看在币基交易中的哈希值所指向的另类币的交易。值得注意的是，这种设计不需要比特币做任何改变，但是需要另类币能够兼容比特币，并且允许共同挖矿。

如果另类币支持共同挖矿，那么我们希望很多比特币的矿工也参与进来，因为这不需要花任何额外的哈希算力。只需要增加少量的运算资源去处理区块和交易，以及矿工需要知道和了解这个另类币，就能去花费精力来挖矿了。假如 25% 的比特币矿工的哈希算力同时在挖另类币的矿，这说明，平均 25% 的比特币含有指向另类币的指针，也就意味着，在另类币体系里，每隔 40 分钟才能产生一个新的另类币。而更糟糕的是，当另类币还在自我发展，并且只有小部分的比特币矿工参与的时候，产生一个新区块需要几个小时甚至几天，这种局面实在让人无法接受。

有没有办法确保参与共同挖矿的另类币的区块，能按照稳定的速度产生？或者说，我们是否可以设定区块产生的速度或高或低，但与比特币中多少比例

的人参与共同挖矿无关？答案是肯定的。奥妙在于，虽然另类币的挖矿任务和比特币一样，但是挖矿的目标不同。另类币体系计算的目标和困难程度和比特币体系中的目标和困难程度都没有关系。就如比特币可以调整其计算目标使每个区块按平均每分钟产生10个的速度一样，另类币也可以调整自己的目标使区块在另类币体系也以每10分钟或其他固定值产生一个。

这意味着，另类币的目标值要远远小于比特币的目标值。部分，甚至是大部分另类币的区块将不会被有效的比特币区块的指针指引到。但是这并不会带来问题，你只需要把比特币区块链和另类币区块链看成是两个平行并列的数据链，只是偶尔有从比特币指向另类币的指针，详见图10.5所示。在图示的例子中，60%的比特币矿工同时也挖另类币的矿，另类币大约5分钟产生一个。这意味着另类币的挖矿难度系数是比特币的60% ×5/10 =30% 。图中40%的比特币区块没有包含指向另类币的哈希指针。

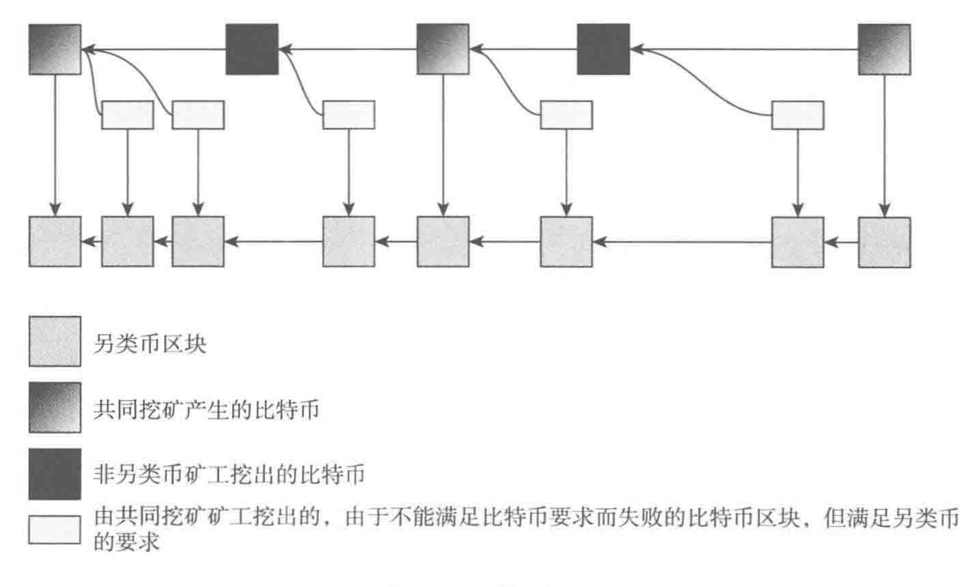


图10.5　共同挖矿

注：图中显示了比特币和另类币的区块链，以及它们之间的相互作用。

相反，每个有效的另类币都是比特币挖矿的结果，但是在所有满足另类币的挖矿算法结果中，只有30%能达到比特币的要求。对于另外70%满足另类币

要求却无法满足比特币要求的区块，另类币的网络需要验证这些区块是否真的符合解决挖矿的谜题。最直接的方法是向比特币相邻区块和另类币区块进行广播。更聪明的方法，是只广播比特币相邻区块的标题部分和比特币区域中包含币基交易的二进制证明。

虽然很罕见，另类币的谜题也可能比比特币更难。由于大部分另类币希望产生区块的速度要快于每10分钟一个，这种情况不常见。当然如果希望放慢速度的话，也很容易做到。这种情况下，你就会看到有些矿工挖到比特币，希望这些也能成为另类币，但是部分比特币区块，在另类币网络中，由于达不到更高的难度要求而被拒绝。

最后需要指出，任意数量的另类币都可以同时和比特币共同挖矿，每个矿工都可以自由选择任意另类币共同挖矿。在这样多种另类币组合共同挖矿的情况下，币基交易的输入脚本本身就是一个指向多种另类币的二叉哈希树结构。注意这种结构的复杂性，因为确认包含另类币交易需要确认以下几点：（1）二叉哈希树包含这个另类币的证明；（2）二叉哈希树证明包含币基交易输入脚本，而且里面包含另类币的哈希值；（3）二叉哈希树证明比特币区块或者附近区块有币基交易的输入脚本。

共同挖矿和安全

共同挖矿是一把双刃剑。正如本书讨论过的，它可以使一个另类币更易实现自我增长的循环发展，通过增加总算力从而提高其抗攻击能力。这种情况下，想通过购买算力去破坏另类币的恶意竞争对手就需要付出巨大的前期投资。

另一方面，有人可能会认为这是一个安全假象。因为恶意竞争对手可以通过共同挖矿来产生比特币，收回一部分前期投资，同时，使攻击另类币的边际成本变得很小。把恶意竞争者看成规模很大的比特币矿工，也许更易于理解。事实上，前文提到的夭折的盘旋币，就是允许共同挖矿的。攻击者矿池 Eligius 和参与攻击者并不需要停止比特币挖矿就可以展开攻击。事实上，矿池的参与者甚至都不知道他们的计算资源被用于攻击另类币。

另类币的挖矿谜题趋势

截至2015年，无论是否允许共同挖矿，很少有另类币使用和比特币一样的SHA－256开采难题。这表明，SHA－256算法被认为是有安全隐患的。Scrypt算法是更受欢迎的选择，因为它使得比特币的ASIC在挖矿或者攻击另类币上变得毫无用处。不过，用于莱特币挖矿所制造的Scrypt ASIC矿机可用于攻击它们。

站在一个理性矿工的角度，当我们思考是否会共同挖矿时，我们会发现共同挖矿有很多安全问题。本书之前曾简略地谈到，只有当期望收益大于期望成本时，挖矿才有意义。对于比特币挖矿来说，成本主要是计算哈希值。但是对比特币矿工来说，决定是否和另类币共同挖矿并不会对哈希计算成本产生影响。额外的成本来自其他方面：计算、带宽、用于验证另类币交易的存储空间，以及需要使软件实时更新及在另类币出现硬分叉或者软分叉时，做出非正式决定。

这样的推理引出两个有洞察力的观点。第一，共同挖矿有很强的规模效应，因为所有的矿工所花费的成本相同，不管其哈希算力有多大。这与比特币有明显差异，因为在比特币体系中，成本和哈希算力成正比。对于小的低价值另类币，由于低的哈希算力，一个小的独立矿工公开挖矿的成本超过了回报，因此其无法获利。截至2015年，通过挖另类币获得的收入只占了比特币收入的很小一部分。这预示着，与比特币体系比，共同挖矿的另类币将会更具中心化。

有预测指出，大部分矿工会选择外包来对他们的交易进行验证。另类币规模越小，矿工就越有动力去找外包。最简单的办法就是加入比特币矿池。因为矿池通常替代矿工进行运算。矿池管理员验证比特币和另类币区块交易，收集添加包含另类币的比特币区块。矿工只需专注于解决挖矿谜题并找出需要的数值。这个预测与实际非常贴近。比如，G池（GHash. IO），曾经最大的比特币挖矿池，同时允许对域名币、IX币（IXCoin）和Dev币（DevCoin）共同挖矿。这些也同时成为最受欢迎的共同挖矿另类币。

第二个观点，从经济学的角度讲，也许更加让人担心安全性而不是挖矿能

力太集中。如果矿工的主要成本是工作量证明，在这种模式设计下，矿工是无法作弊的。在哈希函数的安全性保障下，挖矿没有捷径，并且其他矿工很容易并且也愿意去验证工作量证明。但是如果主要成本变成交易验证时，以上两个假设就不成立了。矿工倾向于假设他们收到的交易都是有效的，并不对这个交易做任何其他验证。而且，矿工如果要去验证一个区块及其交易，其工作量就和挖矿一样。因此，可以预期对于小的共同矿工，他们有动机跳过验证环节。由于存在不验证的矿工，攻击变得更加容易，因为一个恶意的矿工可以创造一个区块，让其他矿工对哪条是最长的有效区块链产生争议。

简而言之，共同挖矿在解决一个安全问题的同时，却也产生其他多个问题，部分原因是共同挖矿和单独挖矿在经济收益上有重大差别。总体来说，考虑到挖矿攻击，共同挖矿对一个新的另类币是否是一个好主意还很难说清。

10.5 不可分割的交叉链互换

在比特币体系里，在不同个体或群体之间，达成一项交换货币或资产的交易很直接。这就是第 6 章里谈到的合币的原理。合币也可以用来交易智能资产，第 9 章简要提到，第 11 章会再进一步讨论。本章前面谈到的在域名币中出售域名也是基于同样的原理。

但是前面所有的例子中，即便涉及不同的资产，交易也都是限制在单一的区块链里。一般来说，一个另类币的交易和另外的其他另类币的交易历史没有任何关系也无法相互参考，这是一个基本的无法跨越的限制。那么，是否有其他办法可以互换不同的货币？比如，如果爱丽丝想卖掉 a 个另类币给鲍勃，换得鲍勃的 b 个比特币，他们可以把这项交易做成是单一且无法分割的形式吗？初看起来好像不太可能，因为无法强迫不同体系的区块链同时发生相关的交易。如果其中一个人，假设是爱丽丝，先执行交易，有什么办法可以阻止鲍勃不遵守承诺呢？

有个聪明的办法可以做到，这用到了密码学的承诺和锁定时间存储，这是

两个我们已经讨论过的技术。图 10.6 描绘了这个协议。暂时先假设两个区块链里的区块是按固定步骤轮流产生的，每个时间单位产生一个区块。T 代表协议流程的开始时间。

1 爱丽丝创建如下 a 个另类币的可以退还存款：
  1.1 爱丽丝创建一个随机的字符串 x，计算哈希值 h = H（x）
  1.2 爱丽丝创建如下图所示存储 A 区块，但是并不公开
  1.3 爱丽丝创建再融资 A 区块，让鲍勃签名
  1.4 一旦鲍勃在再融资 A 区块签名，爱丽丝公开存储 A 区块（但是还没有公开再融资 A 区块）
2 鲍勃创建如下可以退还的存款 b 比特币：
  2.1 鲍勃创建如下图所示存储 B 区块，但是并不公开
  2.2 鲍勃创建再融资 B 区块，让爱丽丝签名
  2.3 一旦爱丽丝在再融资 B 区块签名，鲍勃公开存储 B 区块（但是还没有公开再融资 B 区块）
3 情景 1：爱丽丝按照计划完成兑换
  3.1 在 T_1 爱丽丝索回比特币，把 x 给鲍勃（和其他所有人）
  3.2 在 T_2 索回另类币
 情景 2：爱丽丝改变主意，不要比特币[①]，也不让鲍勃知道 x 值
  3.1 在 T_1 鲍勃索回他的比特币
  3.2 在 T_2 爱丽丝索回她的另类币

存储A[另类币区块]	再融资A[另类币区块]
输入：爱丽丝的价值为a的货币 输出脚本：可以通过签名A以及签名B一起兑现或者通过签名B和xs.t.H(x)=<h>一起兑现	输入：存储A 输出：地址A 定时：T_2 输入脚本：签名A，签名B

存储B[比特币区块链]	再融资B[比特币区块]
输入：鲍勃的价值为b的货币 输出脚本：可以通过签名A以及签名B一起兑现或者通过签名A和xs.t.H(x)=<h>一起兑现	输入：存储B 输出：地址B 定时：T_1 输入脚本：签名A，签名B

图 10.6 不可分割的交叉链互换协议

① 原作者写的是另类币，应该是笔误。——译者注

第一步，爱丽丝存储 a 价值的另类币［这里的存意味着把货币发给输出脚本（ScriptPubkey），并在里面注明只有两种情况可以使用这笔货币］。这个存款只有通过以下两种方法可以取得此 a 价值另类币：第一，如果爱丽丝和鲍勃两个人都同意，他们可以取回。事实上，爱丽丝只有在鲍勃签署回款交易后，她才公开这个存款。这样就保证如果两个时间单位过去后，存款还没有被领取，她可以赎回她的存款。

另外一个办法是在任何时候，提供鲍勃的签名和 x 的值，通过 x 的值去开启哈希函数的承诺 h。注意，把 <h> 写在存储 A 的区块里来表明爱丽丝特意把 h 写入输出脚本。因为只有爱丽丝知道 x，所以在最后阶段，任何单独一方无法索取存款。这个方法就是，当且仅当爱丽丝拿到比特币，鲍勃才知道 x 的地址，他才能索取另类币。

第二步大体是第一步的反向过程。鲍勃存 b 单位比特币，这些比特币只能在两种情况下才能被取走。很重要的区别是，鲍勃并不需要创建一个新的谜题。相反，他用相同的哈希值 h（把这个 h 值从存储 A 区块简单复制到存储 B 区块）。哈希值 h 就是链接两个区块链的钥匙。

这时，爱丽丝有主动权，她可以临时变卦。如果在 T_1 时，爱丽丝还没有表示要给鲍勃 x 值，鲍勃可以简单地取回他的存款退出交易。爱丽丝的另一选择是在 T_1 之前取走鲍勃的比特币，但是她必须创建并广播输入脚本，里面含有 x 值。鲍勃看到这个广播就可以用 x 值去领取爱丽丝的另类币，兑换的交易完成。

注意，如果爱丽丝稍微晚点领取鲍勃的比特币（在 T_1 之后但是在 T_2 之前），鲍勃可能同时拿走两笔存款。类似地，如果爱丽丝及时拿走鲍勃的比特币，但是鲍勃等太久还没取走爱丽丝的，那么爱丽丝也可以把两个都拿走。但这不是问题，只要保证双方无法在协议上欺骗对方就可以，自己的疏忽或者故意怠慢不是系统考虑的范围。

最后，区块在比特币和另类币中，并不是按照固定时间产生的。这种情况会造成一些混乱，特别是两个区块链没有协调一致时。假设两个区块链各自平均 10 分钟产生一个区块。以 1 小时为时间单位，也就是说，需要 T_1 至少为现在另类币区块 +12，T_2 至少为现在比特币区块 +6，也许能带来更大安全边际。

遗憾的是，存在很小的可能性，12个另类币区块已经找到，但后面6个比特币还没有找到。这时，爱丽丝可以索取两个存款。可以通过增加时间单位来降低可能性，但是会牺牲速度。

这是一个清晰明了的协议，但是截至2015年，还没有人用到。相反，所有加密货币都是在传统的中心化的交易系统里交易。造成这种现象有很多原因：第一是该协议的复杂、不便和缓慢；第二，这个协议可以防止偷盗货币，但是不能防范服务性攻击。有人或许以诱人的兑换价格作为广告，但是在协议原型的第一步或第二步就反悔退出，这浪费了每个人的时间。为了减缓这种情况，也为了集合并匹配大家的需求，可能需要一个中心化的交易平台（机制），即使如此，也不能完全相信它不会偷你的货币。这种情况进一步降低了该协议原型的使用范围。

10.6 侧链——基于比特币的另类币

本章前面部分探讨过给现有比特币所有者配置新的另类币的两种方法：或者要求用户把比特币销毁从而得到另类币，或者简单地把另类币发给现有比特币所有者，这些所有者必须拥有还没有用掉的比特币。正如我们看到的，任何一种方式都不需要另类币的价格盯住比特币。没有这种汇率锁定机制，在发展初期，另类币的价格会变化很大。**侧链**（sidechains）的目的就是避免另类币价格变化太大，因为价格的波动太大会导致很多问题，也会使另类币分心乏术，无法真正专注于技术上的竞争。

下面介绍使另类币的价格以固定汇率的形式盯住比特币的相关技术。首先，所有者必须把所拥有的一定数量的比特币放入托管账户，这样才能创造出一个单位的另类币（或者固定单位的另类币），这样所有者才可以在另类币区块链上正常使用另类币。最后，所有者必须能够销毁自己拥有的另类币，从而取回之前存在托管账户上的比特币。这种构建像零币，通过托管基础币而创造零币。区别在于，需要在两个不同的区块链里进行上述操作。

遗憾的是，据我们了解，由于比特币的交易无法被其他区块链的事件所影响，目前还未找到可以不改动比特币而达到这种效果的方法。截至目前，比特币的脚本还没有强大到可以确认整个单独的区块链。好消息是，我们可以通过相对实用一点的软分叉来修改比特币，这也是侧链的原理。侧链的愿景是，将比特币作为储备货币，打造多种蓬勃发展、快速创新和实验的另类币。截至2015年，侧链还只是一个提案。但是比特币社区正在积极参与这个提案，目前已取得一些实质性的进展。侧链的提案还处于变化之中，所以为了便于学习和理解，我们适当简化了一些细节。

扩展比特币的功能，使之能够使侧链兑换成比特币，最显而易见但不太实用的办法是：把所有侧链的规则，包括验证所有侧链的交易和检查侧链的工作量证明，都包含在比特币体系里。这个方法不实用是因为这样会使比特币扩展出来的程序过于复杂，验证比特币的节点会非常困难。而且，链接上的侧链越多，复杂度和困难度就越大。

SPV 技巧

可以使用 SPV 证明技巧来避免这种复杂局面。在第 3 章中，我们曾提到简单付款验证（Simple Payment Verification，简称 SPV）。SPV 可用于小的客户端，比如手机上的比特币应用程序（APP）。SPV 节点不需要对其不感兴趣的交易做验证，它们只校验区块的标题。SPV 客户只看他们感兴趣的交易，并确信是在最长的区块链内，并不担心该链是否是最长的有效链。因为他们假定矿工在创建该区块链并花精力去挖矿之前，已经验证过里面的交易了。

也许，可以扩展比特币的脚本让它能验证侧链里某些特殊的交易（比如销毁一个侧链币的交易）。在比特币里使用这种延展命令的节点，仍然会全面验证比特币的区块链，但是在侧链里，可能只会验证相对轻量级的 SPV。

对一个交易提出异议

这种方法要好一些，但仍不完美。即使做最简化的验证，比特币的节点仍然要链接到侧链的点对点网络（每个链接上比特币的侧链都需要如此），并且追

踪所有侧链区块的标题用于决定侧链最长的分支。最终我们想要的是：当一个交易要把侧链的货币转化成为比特币时，它本身就包含比特币节点需要的用于验证其合法性的所有信息，也就是说，验证特定的侧链是真实发生的。这就是SPV 证明的定义。

这里介绍一种可行的办法，唯一的缺憾是这个侧链的组成部分还在进一步研究中。为了在比特币里可以对照到侧链，用户必须证明：（1）侧链区块里包含侧链交易；（2）侧链的标题表明这个区块已经接受过一定次数的认证，这意味着一定数目的工作量证明。比特币会验证这些证明，但是不会去验证这个区块头部展示的链是最长的。相反，比特币会等一定的时间，比如1～2天，让其他用户去找证据证明，第二步所指向的区块标题并不在最长分支上。一旦在特定时间范围内出现这样的证据，比特币体系中，接受该侧链交易的区块将会被认定为无效。

隐含的逻辑是：如果一个 SPV 证明已经可以确定，该交易不在最长分支上致使其不应该被认可，那么应该有一些侧链的用户会因认可这个交易而遭受损失。这些可能遭受损失的用户，有动力去辩驳 SPV 证明。如果没有用户遭受损失（也许是有一个分支，或者重组侧链，而且该交易也恰好在别的分支上），那接受这个证明也无妨。

一般来说，系统这样设计，对侧链问题并非毫无漏洞，系统也不会阻止你自己搬石头砸自己的脚。如果你把比特币转成有加密隐患的侧链，其他人也许能偷走你的侧链币然后再转成比特币。或者，在侧链的挖矿，也许会因为侧链漏洞而全部崩溃，导致对应的比特币也被偷。但是可以肯定的是，侧链的问题不会毁掉比特币；具体地说，不管侧链有多少漏洞，所有者都无法在侧链上兑现两次同一货币，也就是说侧链不允许比特币挖矿。

通过权益证明精简 SPV 证明的案例

还有一个障碍需要跨越。有些侧链生成区块的速度很快，也许每几秒钟就能产生一个区块。这种情况下，对比特币节点来说，单单验证 SPV 证明就已经负担很重了。这时，可以用一个比较聪明的统计学方法，大幅减少对 N 个区块的验证，也就是大幅减少 O（N）的认证次数。

原理如下：当验证深藏在区块链中的一个区块，其实是在验证每个建立在这个区块上的所有的区块都符合**目标困难度**（target difficulty），即满足哈希值 < 目标值。这些区块的哈希值均匀地分布在（0，目标值）的区域，从统计学角度看，这意味着大约 25% 的区块可以满足哈希值 < 目标值/4. 事实上，寻找 N/4 个区块满足哈希值 < 目标值/4 的工作量和计算 N 个区块满足哈希值 < 目标值的工作量一样。这个数字 4 并无特别，我们可以用任何数代替。

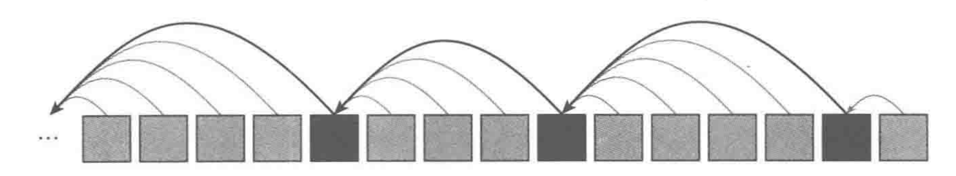


图 10.7　工作证明跳表（proof-of-work skip list）[①]

注：区块包含指向前面一个区块的指针和指向最近的满足哈希值 < 目标/4 的指针。这个原理可以重复运用，比如一个第三层级的指针指向满足哈希值 < 目标值/16[②]，以此类推。

这就意味着如果找到某种方法可以知道哪些区块满足哈希值 < 目标值/4，仅验证这些区块（或者区块的头部）就可以使用 1/4 的工作量完成全部任务。如何找到哪些区块满足哈希值 < 目标值/4 呢？其实答案在区块本身。图 10.7 显示，每个区块包含指向前面一个区块的指针，以及指向最近的满足哈希值 < 目标/4 的指针。

可以压低目标值到多小？是否可以选择很大的数，让目标值变得非常小？答案是否定的。这种方式的原理就像矿池，却是反方向的操作。在矿池里，矿池管理员验证大家的份额，也就是验证这些难度系数低（比较高的目标值）的区块。矿工找到比区块更多的份额，所以矿池管理员必须多做一道验证程序的工作。这样做的好处就是，能够比较精准地估计矿工的哈希算力——估计值的方差较小。

我们来看相反的交易。随着估算建造整个区块链的工作的减少，估算值就

① 跳表是一种随机化的数据结构，目前开源软件 Redis 和 LevelDB 都有用到它。——译者注

② 16 是 2 的 1+3 次方。——译者注

有很大的方差。例如，假设 N =4，在没有使用跳表的方案下，会检测到有 4 个区块满足哈希值 < 目标值。如果一个恶意的竞争对手要欺骗我们，他需要花 4 倍于我们找到一个区块的平均工作量才能办到。

假设这个竞争对手只做了一半的工作。可以算出，竞争对手有 14% 的机会能找到 4 个区块满足哈希值 < 目标值。相反，在跳表方案下用 4 作为倍数，竞争对手的任务变成需要找到一个区块，满足哈希值 < 目标值/4。在这种情况下，懒惰的只做了一半工作的竞争对手，却有 40% 的机会骗过我们，而不仅仅是 14%。

10.7　以太坊和智能合约

我们已经介绍了几种用比特币的脚本语言写出有趣的应用的方法，如有托管功能的支付交易。我们也看到比特币脚本语言的一些瓶颈：只有很少的指令，并不符合图灵计算的标准。因此，很多新的另类币增添与应用程序相关的特殊功能。域名币是第一个尝试，后来又有许多加密货币，类似于比特币但是支持赌博、股票发行、市场预测等。

设想我们不需要为了每个应用程序，每次都建设一套新的系统，而是创造出一个加密货币系统，以支持所有未来可以想象到的任何应用。这就是所谓的图灵完备——据我们理解，满足图灵完备标准的编程语言，可以让你写出图灵机可以完成的任意功能，它可计算的函数和图灵机可计算的函数是完全相同的。因此，每一图灵完备的编程语言——包括我们熟悉的 Java、Python 和 Lisp——都是图灵等价的。如果不考虑实际中的简单性和表现，图灵完备是我们在编程语言有关表达能力上需要的最好的性质。

从某种程度上讲，今天加密货币的发展使人回想起 20 世纪 40 年代早期计算机发展的时代：在第二次世界大战时，建造大量的复杂的只有某种特定功能的计算器（比如用于暴力破解密码的机器和海军用于确定发射弹道轨道的机器），这些工作促使研究专家致力于建造第一个可重复编程的通用计算机。任何可预见的应用程序都可以使用该计算机（见图 10.8）。

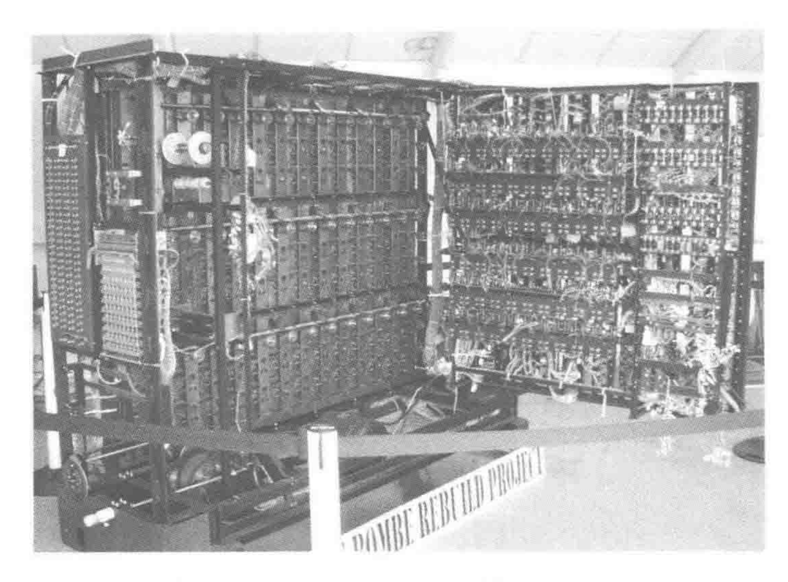


图 10.8　在布莱切利园（Bletchley Park）博物馆重建的炸弹机（Bombe）

注：炸弹机是一个由阿兰·图灵（Alan Turing）设计的特殊功能的高级计算机，用于破解德国的英格玛（Enigma）密码。通用计算机取代类似炸弹机的精巧装备，以太币能否也能像这样取代特殊功能的另类币呢?

以太币（Ethereum）是一种雄心勃勃的另类币，致力于提供一种满足图灵计算要求的可编程语言，用这种语言可以编写脚本或者合约。虽然有其他方案可以做到这一点，但是以太坊无疑是最引人注目的：它使用了几个创新的技术，成功地完成了众筹，在几个月内筹资两千万美元，并且采用激进的参数，比如使用较短的产生区块的时间参数。以太坊系统本身很复杂，需要再编写一本新的教材才能完整阐述，本节只做简要讨论。

智能合约编程模式

智能合约最初是用来指使用计算机系统（或者其他自动化方式）来执行合约。例如，你可以把自动售货机看成一个销售商品的智能合约，执行的就是你和机器主人之间关于如何购买一个糖果的合约。

在以太坊体系，一个合约就是一个存在区块链里的程序。任何人支付一点费用，就可以用特定的操作将他的程序上传，建立一个以太坊合约。这个合约

是用字节码（bytecode）写的，可以被特殊的以太坊专用虚拟机（Ethereum-specific Virtual Machine，简称 EVM）执行。一旦合约上传，便永远存在在区块链里。智能合约有它自己的资金账户，其他用户可以调用程序里面开放的应用程序编程接口（API），合约可以收发款项。

一个简单的例子：以太坊中的域名币

我们说以太坊可以用来执行任何特定应用的另类币功能。举个简单的例子，我们可以展示使用一个简单的以太坊合约，来构建出域名币形式的功能。

图 10.9 所示就是一个构建的案例。它是以“稳健”语言（Solidity）编写，“稳健”是以太坊里用于定义合约的高级编程语言。这个合约产生一个原始的域名/数值（name/value）储存配对或者注册名。名字永远连着数字。这个合约定义了一个数字变量——注册表（registryTable），里面有 32 比特长字节和公开密钥的配对关系。初始时期，每个字节都对应着空地址 0×0000000…000。这个合约同时定义了单一入口点，叫“用户名称”（claimName）。这个入口点只接受名字参数。首先，这个合约确认调用这个合约的人已经支付了至少 10 个 wei。wei 是在以太坊里最小的货币单位。如果没有支付 10wei，合约自动终止并发出错误信号（“throw”的编程命令就做这些）。如果足够的 wei 已经发出而且这个名字还没有被注册，那么这个名字就和调用的地址永久地联系在一起。

```
contract NameRegistry {
      mapping(bytes32 => address) public registryTable;
      function claimName(bytes32 name) {
            if (msg.value < 10) {
                  throw;
            }
            if (registryTable[name] == 0) {
                  registryTable[name] = msg.sender;
            }
      }
}
```

图 10.9　一个用于实现域名注册功能的简单以太坊智能合约

以上就是这个8行代码的合约能做的事。我们还可以多花点时间，把其他域名币有的功能都在这个合约里实现。比如，可以存储拥有者地址以外的信息，通过存储上次更新时间来要求域名的主人定期地重新注册，并且允许其他用户拥有长期不更新的域名权。

我们还可以加第二个功能，允许钱回撤。按照初始的代码设计，钱只能不断堆积在合约里，也就意味着从流通中消失。当然，在可以回撤钱的程序里，最好能设定，调用回撤的是合约的主人。任何人在以太坊都可以调用任何方程，但是用户是指定的，所以能确认谁是真正调用方程的人。

燃料、激励和安全

和比特币不一样的是，以太坊支持循环语句，虽然第一个例子里并不需要循环。循环语句一听就容易让人产生警觉，因为有循环的地方就会有无限死循环。从根本上说，以太坊合约有可能因为种种原因而无限循环。计算机领域里一个著名的研究结果（难以判断的终止问题）证明，不存在任何算法，可以根据源代码去判断一个程序是否可以无限运行下去。因此，我们如何防止合约无限运行下去呢？

更进一步讲，即使条约不会无限运行，也需要某种方式来限制它不会运行太久。以太坊体系通过一种称作"燃料费用"的机制来实现这一点。简单地说，每执行一条虚拟机器的指令需要花费一小部分的成本费用，我们称之为"燃料费用"。不同的操作花费不同。基本的像加减操作只花费1单位燃料费用，而计算SHA-3哈希值（内置函数）需要20单位燃料费用，在永久存储器上写256比特长的字符需要100单位燃料费用。每笔交易也需要先支付21 000单位燃料费用。你可以把以太坊体系想象成超级折扣的航空公司。机票只是你支付乘飞机的费用，任何其他需求都要多付钱。完整操作清单和固定的燃料费用都可以从以太坊里找到。任何清单和费用的变动都需要以太坊产生一个硬分叉，这和比特币脚本语言的语义改变一样。

燃料费用是通过以太坊体系内部被称为以太（ether）的货币来支付的。它只是在用来支付合约操作的时候才叫燃料费用。每笔交易都规定了燃料的价格，

也就是说，每份燃料需要多少以太。燃料费用就像比特币的交易费，矿工可以自由公布交易的燃料费用，每个矿工都可以独立地决定收费方式。这样会得出一个反映市场供求关系的燃料市场价格。2016年年初，虽然以太坊网络体系还是属于实验阶段，市场已经默认50 gigawei为1单位价格。50 gigawei等于5×10^{-8}以太，根据以太币和比特币2016年1月的汇兑比例，这也就是大约3×10^{-10}比特币。

每次调用之前，必须设定燃料费用的最高限，也就是愿意支付价格的最大值。当达到这个值（燃料用完了），程序就会终止，发生的所有程序状态的变化就会被重新设置到原始状态，但是矿工还是保留燃料。由此可见，不要用完燃料，这一点非常重要。

燃料的使用要求，意味着以太坊不适合很耗费资源的计算。以太坊系统未被设计成像云计算那样的服务，即支付一定的费用让云服务完成自己无法做到的计算。像亚马逊的弹性计算云或者微软云计算平台，提供划算百万倍的计算量。另一方面，以太坊更加适合创建安全逻辑协议。本质上来说，以太坊提供了一种两个或者多个匿名交易者可以信赖的服务系统。

以太坊上区块链的安全还没有像比特币一样完善。理论上，以太坊比较复杂，也比较难以用数学推理来论证。实际上，以太坊才刚刚开始发展，其安全性还没有像比特币一样经过很多考验。尤其是，担心处理交易的成本会让类似比特币的激励机制失效，我们在共同挖矿的分析中讨论过存在这种担心的情况。当交易成本占矿工的总成本的比重不再能忽略不计的时候，大的矿工有明显优势，因为成本和哈希算力相互独立。更重要的是，燃料只支付给最初包括该交易的区块的挖工。但是所有的在这之上建立区块的矿工都必须验证该区块，却得不到任何报酬。这意味着，他们将有动力去跳过该验证。正如之前所看到的，这种情况不利于区块链体系的健康发展。

第二个例子：以太坊体系中的国际象棋

我们还没涉及以太坊中新功能如何运用，所以让我们看第二个案例。假设爱丽丝和鲍勃下国际象棋，赌注是一定数额的金钱。唯一的问题是爱丽丝和鲍

勃生活在不同的国家，他们都不相信对方输了会支付赌注。这个问题可以用以太坊来解决。

爱丽丝写下以太坊程序，这个程序设定了国际象棋的规则并且被上传到以太坊网络。她给这个合约支付一定数量的以太作为赌注。鲍勃可以看到这个合约，如果他答应接受挑战，他把他的赌注支付给这个合约，就等于开始了这个游戏。鲍勃在接受挑战之前应该确认，这个合约是准确无误地遵守了国际象棋的规则，并且最后会把所有赌注支付给获胜者。

一旦双方都支付了赌注，假设他们约定下同样的赌注，合约会检查双方的赌注是否相等。这时候，游戏就开始了。任何一方除非赢了游戏，否则无法从合约里取出钱来。其他人在任何情况下也无法取得这笔钱。

爱丽丝和鲍勃轮流把自己的下棋步骤发给这个合约。这个合约也会检查轮到谁下确保指令是由爱丽丝或者鲍勃发出，而不是其他人。大家是否还记得调用者需要在每个操作（促使合约执行一个动作）上签名，因为合约可以根据签名确认调用者。合约也会根据国际象棋的规则校验双方的步骤。如果一方试图把兵移动3格，合约会拒绝该步骤。

到最后游戏结束。合约在每一步都会检测是否有一方被将军，或者双方打平，或者满足其他打平的条件。玩家也可以发送投降的指令。当游戏结束时，合约终止，并把所有的钱支付给获胜者，或者平局下平分赌注。

从概念上看，这是一个以太坊的简单应用，但是有很多微妙的地方值得探讨。如果一方快输了他就放弃了？合约应该设定一个机制，如果一方在规定的时间没有提交有效的下一步，钱就支付给另一方。

哪个玩家先走呢？白方先走的话，白方就拥有微小的优势。因此，双方都想做白方。这就碰到了以太坊合约的一个难题：没有内置的随机源。之所以是一个难题，是因为随机数发生器需要所有矿工的检验（因为他们需要检验合约是否正确地执行），但是这些随机数对任何人来说都是不可预测的（否则的话，玩家也许就因为不能先走而拒绝参加这个游戏）。

随机数“信号塔”（randomness beacons）可以解决这个问题。正如9.4节讨论的，在双方都加入游戏后，合约计算区块链下一个区块的哈希值。对这个特

定的游戏应用而言，这个问题比较容易解决，因为只要让爱丽丝和鲍勃双方确信决定谁先谁后是随机的，这样就满足要求，而不需要向所有人证明。所以他们可以采用9.3节的办法：他们两个同时提交一个随机数的哈希值，并且公开他们的输入值，然后从双方的输入总值算出随机数。实际操作中，以上两种方法都可以使用。

其他应用

下棋也许很有趣，但是真正激动人心的是以太坊在金融领域的应用。我们在课本里讨论的大部分应用，包括市场预计、智能资产、托管支付、微支付渠道和混合服务，都可以在以太坊体系里实现。这些应用都有其细微的区别，但是相对比特币死板的协议，大多数情况下，这些应用都能相对容易地在以太坊体系内完成。

以太坊的状态和账户余额。第3章中，我们讨论了账本的两种方法：基于账户和基于交易。在一个基于交易的账本中，如比特币，区块链只存储交易（加上一些少量的转载标题的设置数据）。为了方便验证交易，比特币的币值是无法分割的，即交易的结果必须整体被消费，可以自己消费，或者如果需要的话，换地址消费。交易实际上是在全球状态表上操作的，这个表称为“未花费交易输出列表”。但是比特币的协议并没有明确规定这个全球状态表。全球状态表的产生纯粹是矿工为了加快验证过程而创造出来的。

另一方面，以太坊则是基于账户的模式。由于以太坊已经存储了合约地址和状态的对照表的数据结构，很自然地也同时存储每个普通地址（或者叫拥有者的地址）的账户余额。这意味着，与非闭环式的交易支付模式必须有输入和输出不同，以太坊存储每个地址的账户余额，这一点，与银行存储每个账户余额的方式类似。

以太坊的数据结构。在第3章，我们提到基于账户的账本需要精心设计的数据结构来存储记录。以太坊就有这样的数据结构。具体来说，每个区块包含每个地址的目前状态（账户余额和交易数）的摘要，同时也包含每个合约的状态（余额和存储空间）。每个合约的存储树结构映射256比特的地址和256比特

的字节。这样可以存储巨量的（$2^{256} \times 256 = 2^{264}$）信息。当然，这只不过是理论上的可能空间，我们不会用到这么大的存储空间。数据结构里面提供的摘要，使验证一个地址有多少余额或者空间变得相对容易。比如，不需要鲍勃从头到尾扫描整个区块链，爱丽丝就可以向鲍勃证明她有多少余额。

此时，比特币用简单的二项梅克尔树的结构可以派得上用场。因为它可以把有效的证明数据存在该区块里（要求矿工确信对于相同的地址，每个树状数据结构都要求该地址相同的状态）。但是我们也希望能够更快地查询地址并且能够有效更新地址的数值。为了达到这个目的，以太坊使用比较复杂的树状结构，叫帕特里夏树（Patricia tree）、前缀树（prefix tree）、字典树（trie）或基数树（radix tree）。每个以太坊区块包含梅克尔－帕特里夏树（Merkle Patricia tree）的树根，它保存每个地址的状态，也包含合约地址。每个合约的状态，包含一个树状数据结构用来保存合约的存储状态。

基于账户账本的另一个不易处理的问题是防止重复攻击。在比特币里，每个交易都使用“未花费交易输出列表”输入，因此，任何相同签名认证过的交易，不可能被重复使用两次。但是在以太坊设计里，需要确保当爱丽丝签下支付给鲍勃 1 以太交易的时候，鲍勃不能一次又一次地对外广播并重复使用这个 1 以太，直到把爱丽丝的账户用光。这样的交易不能重复，因为一旦使用了，爱丽丝的交易计数会增加一次，而这个交易计数是一个全局的状态参数。

总的来说，以太坊使用比比特币更加强大的数据结构来管理它的账本。虽然我们没有深入研究它的数据结构，但我们知道，这个数据结构使得账户、合约，以及交易相关声明的有效验证变成可能。

以太坊项目

以太坊最早于2013 年年末开始讨论，并于2015 年第一次发布，代号先行者（Frontier）。以太坊采用预售的方式，以固定比特币价格公开出售，并把所有的预售款投入以太坊基金会。

和其他另类币相比，以太坊发展比较缓慢，这也反映了以太币是一个比较复杂的系统。与比特币相比，以太坊增加以太坊专用虚拟机（EVM），一个全新

编程模式，一个全新的数据结构。此外，以太坊还对比特币的共识模式做了大的修改。每个区块产生的时间不是10分钟，而是12秒。在以太坊体系里，过时区块的比例高于比特币体系，为了减少过时的区块对系统的影响，以太坊采用另一个叫“精灵”（GHOST）的协议来计算共识分支。同时，以太坊采用不同的工作量证明。目前采用的是一个混合的哈希方程，被设计成只能用记忆体计算。未来以太坊计划转为通过权益证明份额证明的体系。

以太坊呈现出和比特币在设计理念上的巨大差异。以太坊项目由非营利机构主导并且在规划和决策上相对比较集中，它们根据历史经验对以太坊协议进行修改，并且都有一个公开的时间表。按规划，将来也会有硬分叉。而且，所有以太坊合约都要在版本更新前销毁。所以，以太坊还是一个未来会有很多变更的实验性体系。截至2015年，投入大量精力在以太坊上并构建真正有用的应用，现在看来是有点太早。但是以太坊无疑是一个非常有潜力的系统。也许这本书未来的版本将会命名为“以太坊和加密货币技术”。

本章主要讨论了比特币如何成为其他加密货币和另类币的重要组成部分。它们在各方面相互竞争、合作并且相互影响，有些相辅相成，有些是相互阻碍。在未来，可能会出现新的技术，使得在一种区块链可以直接引用另一个区块链的交易。

但截至目前，有些问题仍然悬而未决。另类币会演变并相互合并，最后成为少数另类币主宰的生态系统吗？还是和现在一样多样化？特定功能的另类币会蓬勃发展，还是以以太坊为标志的通用编程平台最后成为主流？比特币和另类币之间的相互影响是有益的吗？每个另类币之间应该相互独立，比如用不同的挖矿谜题而不是去共同挖矿？我们现在无法回答上述这些问题，但我们已经讨论了所有这些你需要理解和评估的重要概念。

延展阅读

侧链白皮书：

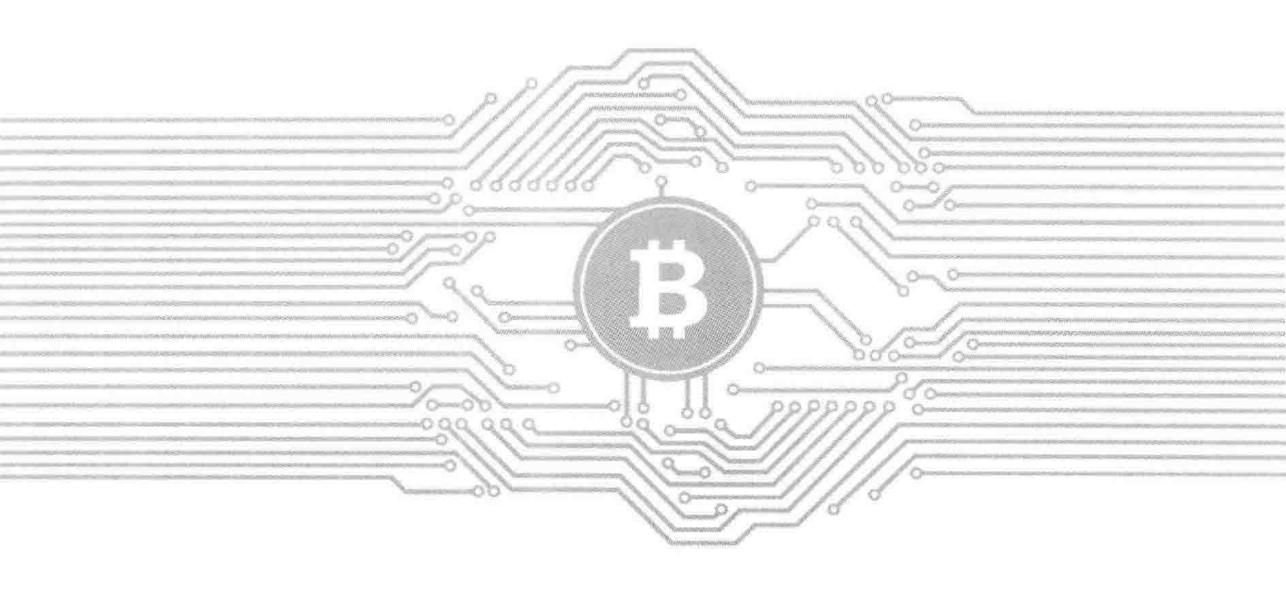
Back，Adam，Matt Corallo，Luke Dashjr，Mark Friedenbach，Gregory Maxwell，

Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. "Enabling Blockchain Innovations with Pegged Sidechains." 2014.
您可以通过如下网址阅读：
https：//blockstream. com/sidechains. pdf.

关于域名币和使用其他用加密币设计域名/价值储存方法的论文：
Kalodner, Harry, Miles Carlsten, Paul Ellenbogen, Joseph Bonneau, and Arvind Narayanan. "An Empirical Study of Namecoin and Lessonsf or Decentralized Namespace Design. Presented at the Workshop on the Economics of Information Security, 2015.
您可以通过如下网址阅读：
http：//randomwalker. info/publications/namespaces. pdf.

以太坊白皮书：
Various authors. "A Next-Generation Smart Contract and Decentralized Application Platform."
您可以通过以下网址阅读：
http：//github. com/ethereum/wiki/wiki/white-paper.

分析以太坊激励机制错配的学术论文：
Luu, Loi, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. "Demystifying Incentives in the Consensus Computer." *Proceeding of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York：ACM, 2015.



第11章

去中心化机构：比特币的未来？



这本书写到这里，我们已经研究了比特币和区块链技术截至2015年的最新状态。本章将讨论的是，比特币在未来会怎么样。我们恪守箴言“绝不预言，特别是预言未来”，所以我们不会声称知道比特币未来会怎么样。因而我们在本章标题使用了问号。

但是我们继续本书中一直坚持的学术探讨风格，用以研究未来的科技。比特币的未来充满了热情和叹为观止的未来科技革命的前景。本章可以写成一个宣言，但我们不准备这么做。我们找出知名的方案，用谨慎冷静的方法对它们分类，并批判地分析它们的相对优势和劣势。

比特币是一个内容丰富的课题，它包含了协议本身及其成为新应用平台的潜在可能性。本章的重点不在于比特币协议的未来。虽然我们意识到有很多影响比特币未来且值得研究的方面，比如，比特币的管理、效率、扩展性和功能集合。

我们会着重讨论比特币如何成功地成为去中心化的货币，它的成功经验能促使我们重新思考其他中心化的机构——比如从事股票、债券、资产产权和其他业务的机构。我们不禁会问，区块链是否能够用来对它们去中心化。我们不仅仅要考虑技术上去中心化的可能性，还要考虑，它是否合乎金融规律并对社会有利?

11.1 区块链作为去中心化的工具

在比特币之前，有很多数字电子货币失败的尝试（本书的前言中提到了许多失败的案例）。比特币与大多数失败的尝试之间，最重要的区别在于去中心化。比特币去中心化的核心创新方法是使用区块链。

在这一节，我们会研究区块链技术如何在货币体系以外的领域里实现去中心化。本章我们会一直重复使用汽车的案例，这辆车的所有权是通过区块链来控制的。这个案例是我们在第 9 章中介绍的智能合约的概念里的具体体现。早在 1990 年年初，尼克·萨博（Nick Szabo）和其他几位创新性提出智能资产和管理智能资产的数字化合约，远远早于比特币。随着区块链的出现，这个想法变得越来越现实和具体。

令人鼓舞的案例

当代的汽车使用两种主要的加锁机制（locking mechanisms）：门上的物理锁和通过电子锁住引擎不让发动的制动器。用户用遥控钥匙可以与汽车无线通信，根据遥控器与车的距离或者用户按下按钮的特定动作，来解锁车门和发动引擎。

为了防止假冒钥匙入侵，车钥匙的解锁机制需要加密。虽然安全专家发现很多最近使用的加锁机制存在很多问题，但是他们是有可能处理好这些问题的。一般来说，这些算法会使用对称钥匙加密技术。在我们这个案例中，用的是基于非对称加密技术的类似椭圆曲线数字签名算法的数字签名技术。

在这个例子里，汽车储存了一份遥控器的公开密钥，用于打开车门和发动引擎。当遥控器请求开门的时候，汽车发送回随机数并要求遥控器用它储存的密钥签名。只有遥控器准确回复签名，车门才会被打开。到目前为止，这个原理和实际中的防盗机制没有什么大的区别。唯一的区别是，我们使用了很深的

加密技术，安装起来比较昂贵。

实现智能

这次设计的智能汽车，是假设用来验证遥控器的公开密钥并不是靠汽车制造商永久地记录在汽车里；而是，智能汽车技术上可以不间断地，无线接收例如比特币一样的区块链上的新区块。当汽车在组装厂组装的时候，遥控器里第一个用户（比如组装厂的经理）的公开密钥通过特殊操作加入区块链。同时，这辆智能汽车也把该特殊操作的ID写入它自带的程序。

核心的思想是，汽车更改所有权的时候——从装配车厂到质量监控室到运输人员到汽车经销商到第一个所有者——也同时更新到区块链，区块链同时授权每一步的转换。值得注意的是，在这个模式下，授权用的遥控器没有跟着车走。每个人或者公司都有一个预先存在的遥控器（或者带着某种有遥控器功能的仪器）。这个遥控器里面有一个唯一的签名密钥。根据区块链的交易，签名密钥被激活或者被取消拥有这辆车的权利。这样的交易以车最新的交易ID为输入，同时设定一个新的公共密钥为输出ID。汽车目前的拥有者需要用私人密钥在这个输出ID上签名。

这种设计和我们在第9章讨论的智能资产相似，除了一个重大的区别：区块链的交易不仅仅表示汽车所有权的变更，它还代表真正的汽车物理拥有权的转移。当汽车通过区块链转移的时候，前车主的遥控器无法工作，新车主的遥控器获得开门和启动引擎的权利。让所有权等同于使用权的技术有着深远的影响，这将促使强有力的去中心化。但是去中心化是否有用，这并不容易看清楚。我们将在11.4节回过头来讨论这个问题。

安全的交易

假设爱丽丝拥有一辆智能汽车想卖给鲍勃，能够数字化地转移汽车控制权会引起几个有趣的可能性。比如，爱丽丝也许正在国外旅行正需要钱来支付旅费，所以要卖掉停在她家后院车库的汽车。只要联上互联网，鲍勃就可以用比特币支付给爱丽丝车钱，爱丽丝可以远程通过区块链把车的所有权转移给鲍勃，

鲍勃就可以开走这辆车。

然而，这样的交易存在一定的风险。如果鲍勃先支付，爱丽丝也许收了钱而不转移车的所有权。如果爱丽丝先转移车的所有权，鲍勃也许不付钱就把车开走了。即使假设爱丽丝在现场，也有可能另一方突然改变主意而撤销交易。这时候让不在场的第三方来调解争议也很困难。

我们之前多次碰到这种问题，包括在合币（第6章）和域名币（第10章）中。解决这类问题要使用同样的原则。只要支付的货币和汽车的拥有权同时存在相同的区块链，爱丽丝和鲍勃就可以产生一个不可分割的交易。这个交易同时转移汽车的所有权和车款。具体地说，这个交易规定两个输入：爱丽丝的所有权和鲍勃的支付款；规定两个输出：归鲍勃的所有权和归爱丽丝的支付款。这个交易需要双方提供输入要素，因此要求双方都要签名。如果只有一方签名，交易就无效。一旦一方签名，交易的细节就无法改变，除非这个签名无效。签过名的交易一旦对整个区块链广播，鲍勃只要等预设的几次确认（一般是6次），就可以拥有这辆车。鲍勃支付给爱丽丝的款项也同时被确认。两个确认是相辅相成，缺一不可的。

细心的读者也许注意到了一个微妙的问题。鲍勃可以收到爱丽丝签名的交易，自己也签名，但是不立即对外广播。等到爱丽丝卖东西的价格变了，鲍勃才把旧交易用原来的价格对外广播。所以为了避免这种问题，比较复杂的不可分割的交易里包括截止时间。过了截止时间后，爱丽丝可以发送输入她控制下的新地址，用来表示撤回她发给鲍勃的已经签过名的交易。

这是本章案例之一。我们将会在本章看到，很多其他案例使用区块链技术促使现实中的各种交易程序去中心化，从而达到不同种类的去中心化状态。其中，不可分割性（atomicity）是绝大多数案例共有的特性。也就是说，交易每一方的交割都是联系在一起的，所以它们都同时发生或者都不发生。不可分割性是区块链以外应用程序领域里重要的安全概念。

11.2 通往区块链融合之路

因为比特币的区块链是专用在货币上的，把它改造为用于表示其他应用是很有挑战性的。在比特币世界里，你会发现有许多人偏爱把比特币或者别的区块链作为去中心化的平台。在本节，我们来分析两种方法。

方法1：直接在比特币基础上

区块链融合自然而然的出发点是比特币。这也是我们在前面11.1节智能汽车例子里用的办法。直接使用比特币的好处就是容易实现——代码容易运行，比特币网络有很强的挖矿能力，共识过程没有瑕疵。然而，我们必须在比特币上做些修改才能用于我们的例子。比如，用于授权比特币交易的加密要等同于用于打开车门的加密。有时候对比特币的修改是不可能的，而且从根本上说，如果你有非常复杂的涉及不同方的合约，用比特币的区块链不一定能足够胜任或者不可分割地执行。为了展示用比特币区块链的危险性，我们研究一下如何构建一些中性的非中介化的应用程序。

首先，我们来研究众筹服务。在2015年，最广泛使用的众筹网站是Kickstarter，它通过一个中心化的网站，连接了创业者和资金提供方。我们欣赏Kickstarter的想法，但希望通过建造一个完全去中心化的替代系统。这个系统需要让创业者能要求投资人捐款，但是在收到一定预先设定数额之前，创业者不能花掉任何一分钱。所有的这些都是没有中介的。

用比特币的技术实现这样的众筹服务，需要创业者创建一个特定输入的交易（输入数可以随着进程而改变）和一个支付给自己的输出，比如支付1 000个比特币（BTC）。这个交易将在潜在的资助者中流传。任何资助者都可以把资助额加在交易的输入上，并且数字化签名他们的输入和总输出。只有到所有输入等于或者大于输出的时候，创业者才能取得这笔交易的所有输入（见图11.1）。因为签名形式有限，我们要用到比特币一些鲜为人知的功能，才能花掉

最后的交易额。虽然这在当今的比特币系统能做到，但是我们必须钻研到比特币里很少人知道的角落。这并不是一个日常见到的标准比特币交易。

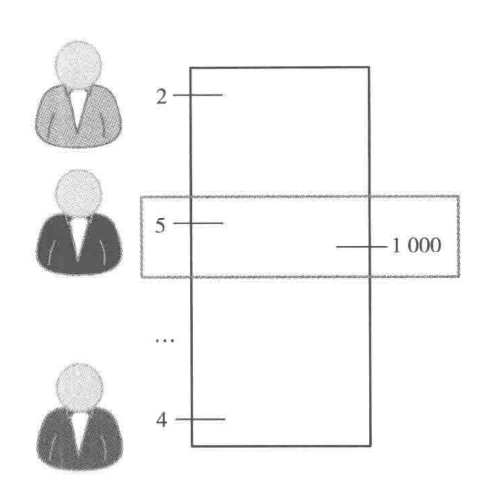


图 11.1　通过比特币众筹

注：由不同资助者发起的多输入的单笔交易。每个资助者签下他们的输入和输出。只有在累积输入数额达到或者超过输出，该交易才有效。

另一个案例：**支付证明费用**（paying for a proof）。这个案例初看起来好像奇怪，但是它有很重要的应用。为了表示清楚，我们假设有个哈希函数 H 和一个大家都知道的数字 y，y 是 H 输入 x 后产生的输出。爱丽丝知道这个 x 的数值，鲍勃为了知道这个 x 值，愿意支付给爱丽丝。广义上讲，H 可以是任何计算程序，鲍勃希望知道他感兴趣的能够产生特定结果的输入值。这个问题的进一步演变是，鲍勃也许愿意支付一定的费用，让这个输入值公开在区块链上。

为了安全地实现这笔交易，我们必须确认交易的不可分割性。爱丽丝只有提交正确的输入才能收到钱。而鲍勃一收到输入必须承担支付责任。记得我们在第 10 章的不可分割的交叉链互换协议中，展示了如何绑定支付和呈现哈希函数输入值。类似的方法可以用在这里。

这些例子显示了直接使用比特币区块链的重要局限性。在每个例子中，我们必须把复杂的真实世界交易编译成比特币的概念。这不一定总能实现。在智能汽车的例子里，我们假设该车用 ECDSA 签名技术来验证汽车的主人。这就允

许我们使用区块链和遥控器相同的公/私密钥来开锁并发动汽车。在众筹例子中，创业者只能拿到他们要求的数额，不能多也不能少。如果资助的金额大于需求的，多出的部分就成为交易费用。最后，在支付证明费用的例子里，如果函数 H 不是比特币语言所支持的哈希函数，那么连接支付和公开数字这种模式就很困难。

如果你不能，或者不想把应用程序强行套入比特币的交易体系，那么可以选择使用附着币，我们在第 9 章讨论过附着币。这样比特币就成为数据存储，因此比特币的脚本语言如何表达就与之无关。这种用附着币的方式，不仅可以构建很多其他应用，还可以使应用透明化。重新回到卖车的例子，如果区块链中对象的颜色是公开的（比如以有颜色的货币来表示），任何人都可以通过检查区块链得知汽车在何时购买，以什么价格，而不用知道买卖双方的真实身份。这种方式在某种情况下很有用。在对它不利的时候，颜色的对象可以不对外公开。

然而，这种附着币有重大缺陷。附着币的用户无法依靠比特币的矿工来验证交易（因为矿工无法了解附着币的交易语言含义）。这意味着所有附着币的用户必须以全网节点的方式运行。SPV 也不可能。只要在构建的时候存在使得共识无法达成的漏洞，附着币就会变得很不稳定。如果两个附着币对一个交易的有效性持不同意见，会导致这个货币形成两个分叉货币。这会导致灾难性的后果。相反，如果由矿工来验证交易，这种不同意见的情况就会很少发生。如果这种情况真的发生了，那么它很快就会引起注意并且很可能解决，而不会导致货币分叉。

从另一方面考虑，无论我们是否用附着币，用比特币的初始范畴外的交易会加重或者“污染”比特币的区块链。在比特币世界，这个问题双方争执不下。我们不选边站，但是我们要指出有一个解决的办法：就像我们在 9.1 节看到的，仅仅把比特币当成时间戳服务，而不是当成数据存储。目前有刚刚起步的服务，提供另外的区块链或者数据存储服务。其中有一个服务是通过比特币区块链做时间戳服务。这就像第 9 章讨论过的时间戳服务，但是多了每 10 分钟一次哈希计算，而不是每周一次在新闻报纸上。用比特币当时间戳只要每区块（或者每

次服务或者协议）一次交易。不完美的地方就是，很难找到像比特币区块链这样容易获得并且广泛复制留存的外部数据存储。同时，一般的非比特币数据存储更加中心化。

总的来说，不管是否使用嵌套技术，比特币的区块链诞生了许多创新的应用。这些通过比特币区块链产生的应用，受到用户和矿工的广泛接受。因此，使用比特币区块链是一个安全且容易实现的选择。

方法2：另类区块链

去中心化的另一个方法是使用另类区块链，也有几个选择方案。最明显的方法就是，使用一个全新的区块链，有自己的规则、功能和货币（也就是另类币）。另一个方案是，使用我们在第10章学过的侧链。这个方案的主要不同在于侧链的货币是1∶1的比例方式与比特币挂钩。有高级脚本语言的侧链可以满足复杂合约的要求，也能做到去中介化。但是，侧链需要对比特币进行修改，这些修改目前到2015年还没有进展。

第三个选择是用已经存在的另类区块链，这个区块链能够支持新的应用程序。截至2015年，我们在第10章讨论过的以太坊体系，是最有潜力的去中心化加密货币应用程序平台。从概念上看，以太坊是去中心化的复杂合约的理想平台。当然，以太坊也面临实际的挑战：至少到2015年，它还没有达到比特币同等水平的成熟度、接受度和挖矿计算力，也没有接受相应水平的安全性考验。无论如何，这是一个令人着迷的去中心化复杂合约的精美实验。未来，以太坊体系或者其他体系有可能蓬勃发展。

11.3 去中心化的模板

我们研究了几种通过区块链达到去中心化的方式。接下来，我们将会建立一套去中心化应该考虑的问题的模板，问题包括去中心化的对象、用什么区块链合适、实体和安全性在去中心化下的重新定义。

去中心化的程度

通过去中介而去中心化

再回想一下我们前文提到的智能汽车，为了帮助理解，我们问一个问题，数字拥有权的转移到底替代了现实生活中的什么步骤?

美国汽车的产权，是由拥有产权证书来确定的。这是产权的中心化形式。产权证书只有机动车管理部门（Department of Motor Vehicles，简称DMV）能识别和实际使用。当出售汽车的时候，卖方交接产权证书给买方，仅仅这样做是不够的。这样的产权转移，必须在DMV注册，DMV在其中央数据库进行更新。通过区块链来转移产权，我们从国家掌控的集中程序转变成不需要任何中介。这就是通过去中介来实现去中心化。

争议的调解：通过竞争去中心化

假设买卖双方对汽车交易有争议。也许卖家出售了一辆以次充好的汽车，买家很生气要退货。在第3章，我们讨论过3选2的多方签名交易。如果除了买卖双方以外还有第三方比如法官或者调停者，这个交易就可以支持托管。这种情况下，买方不需要直接把钱转给卖方，而是转到一个由买卖双方和调停者组成的3选2控制的地址。调停者在另外一方的帮助下可以批准支付或者退款，但是不能偷走这笔钱。

这是一个很好的争端解决机制的开端，但仍有很多细节需要考虑。首先，我们丧失了以前依赖的汽车交易的不可分割性。其次，我们不清楚汽车的拥有权是否可以随着退款而恢复原来的状态。最后，如果汽车的所有权转到3选2的地址，那么遥控钥匙要给谁开车的权利?我们讨论这些的目的不是找出解决这些问题的办法，而是用这个例子来仔细思考调停者的作用。具体而言，我们要比较这种调停模式和传统调停模式的区别。

现实世界里的调停争议如何发生?这需要借助司法体系，一个中心化的、国家控制的，最好需要聘请律师的调停系统。相反，在数字合约下，参与者自

由选择他们想要的调停者。一个调停的私人市场将会蓬勃发展。在这个市场里，不需要按现有的法律规定，各种潜在的中介可以根据公平、效率和成本相互竞争。这中间有几个挑战性的问题：第一个是激励机制，调停者也许会被交易的一方贿赂；第二个是资金在有争议期间是被锁住的；第三，当内部调停机制失效后，因为交易双方都是匿名的，所以很难最后上法庭解决，即使双方可以被识别，数字合约当前也不被法庭认可。

我们的观点是，这不是通过去中介而产生去中心化——我们没有完全摆脱中介，而是让参与者选择他们信任的人。也就是说，通过竞争去中心化。在介于单一规定的中介和完全不需要中介的完全去中介之间，还有很大的空间和可能。就像我们看到的，在这两种极端之间，可以存在一种情况，即有很多中介相互竞争。事实上，我们在第9章讨论去中心化的未来预测市场也是这种情况。不像InTrade① 一个公司占领整个市场，参与者可以从竞争的仲裁人中任意选择他们相信的人来进行关键的市场操作。

能达到什么样的安全程度

我们从这个案例观察到另外一个问题。诊断解决过程的安全性不需要依靠不可分割性。相反，它依靠对调停者的信任。调停者如何变得可信？有很多种办法，最直接的就是通过信誉。与不可分割性通过技术维持安全性不同，信誉是建立在长时间的社会内部机制相互作用的基础上的。

信任

有人在比特币世界里把“信任最小化”或者“没有信任”作为目标。这听起来像是退步。难道我们不是希望建立一个值得信任的可以运作的体系？

信任这个词有不同的解释，因而会引起不同的误解。当爱丽丝借给鲍勃

① 一个在线电子交易平台，对应的在线赌博模式被称为“未来事件交易”。网站用户可以对所有与体育竞赛没有关系的“大事件”下注，包括政治事件、经济形势、娱乐新闻、交通运输、法律现状、流行音乐甚至是天气情况。——译者注

10美元并说信任他，爱丽丝的意思是她觉得鲍勃是个信得过的人，她坚信鲍勃会还钱。在安全术语里，一个可信任的组成部分意味着你不得不依靠的对象。当人们用可信任来描述认证机构的时候，他们觉得如果这些机构都不可信了，网络安全就毫无保障了。

在其他条件都不变的情形下，信任最小化是一个值得追求的目标。我们希望能打造一个安全的系统，这个系统能够尽量减少我们不得不依靠的组成部分。但是当你手里有把铁锤，任何事情看起来都像可以用铁锤来解决的钉子。比特币的支持者经常过度追求去除体系中的可信任部分。一个可信的组成部分不总是坏事。现实世界里存在的信任关系本身也没有问题。去除可信任的部分，可能带来其他隐藏的问题。

我们在本章最后的11.4节会详细阐述这一点。至少到目前为止，我们意识到信任这个词的复杂含义，我们避免信任这个词，而是用安全这个比较明确的词来表示。

在技术解决不了的地方，或者作为技术解决方案的补充，信誉起了一定的作用。然而，信誉也有一些问题。信誉是要和真实身份联系在一起的。如果真实身份不确定或不明朗，信誉就无法发挥作用。比如，一家餐馆收到网络差评后决定关了该餐馆，摆脱坏信誉，然后以原班人马重开一家新的餐馆。在匿名的世界，信誉无法发挥作用。在半匿名环境下，真实身份很容易改变，以信誉为基础的系统会面临很大的挑战。信誉体系也要设法去验证对别人声誉有影响的众多评价。在传统的像Yelp[①]的系统中，商户用真名，用户某种程度上也用真名。然而在半匿名环境，很难精确地区分是假的指控还是真的事实。

我们不再深入讨论其他的安全机制，比如安全硬件设备等。不管是用什么安全机制，最后都会面临一个大的挑战，即没有现实生活中的执法机构。因为没有用真实身份，这些机制里都没有对错误行为的惩罚，争议也无法在法庭上

① Yelp类似于国内的大众点评。——译者注

解决。借钱变得不可能，因为没有执法机构保证借款人会还钱，因此交易经常需要先预存款，这些预存款在争议期间会被锁住。

框架

总结一下本章至今讨论的内容，我们可以通过提 4 个问题，来对任何去中心化的方案进行分析：

1. 去中心化的对象是什么？
2. 去中心化的程度如何？
3. 用什么样的区块链？
4. 采用什么样的安全措施？

只要回答了以上 4 个问题，我们就基本上能精确地了解大多数在比特币世界里见过的基于区块链的去中心化方案。让我考虑以下几个例子（见表 11.1）：

智能资产

我们讨论过的智能资产，去中心化的是资产的所有权和资产的转移。它达到了完全去中介化——完全不需要 DMV 和国家这样的实体。我们谈过它如何使

表 11.1　区块链去中心化的方案特征

特征	方案			
	智能资产	去中心化的预测市场	存储 J	零币
1. 去中心化的对象是什么？	资产所有权和转移	预测市场	文件的存储和调用	混合币种
2. 去中心化的程度如何？	去中介化	竞争和去中介	竞争	去中介
3. 用什么样的区块链？	比特币	另类币	比特币	另类币
4. 采用什么样的安全措施？	不可分割	声誉，不可分割	声誉	不可分割

用比特币的区块链来达到目的。当然你也可以用另类区块链。最后，安全性是通过不可分割方式，即把支付与汽车所有权的转移不可分割地融合在一起。

去中心化的预测市场

在一个中心化的预测市场，中心平台或者交易所提供两种重要的服务：裁决并公布任何赌注的事件，并向参与者出售股权（或者帮助参与者安全地彼此交易）。我们在第9章看到，去中心化的预测市场不需要一个中央权力机构来提供这些功能。它让每个人都可以为一个时间设置一个市场，并通过发送一个简单交易，成为对应市场的仲裁者，这样降低了执行这些操作的准入门槛。因此，中介还是存在的，但用户可以自由选择很多相互竞争的中介。如果用户还不满意，他们自己可以来执行这些操作。另一方面，用户可以直接自动地和其他用户交易股票，所以中央集权这方面的功能被去中介化了。去中心化的预测市场要求比特币没有的新功能，所以最好通过一个定制的另类币的区块链来实现。

存储J

存储J（StroJ）是格雷戈·麦斯威尔（Greg Maxwell）提出的，用于文件存储和调用。它已经有好多更新版本，但我们只讨论简单版本。大体上看，存储J设置了一个存在云端的“代理”，它被设计过，能自己做出一些特定决定。比如，为了有计算资源，它可以租用云端算力和存储空间。另一个提供给用户的功能是让客户将文件存放一段时间，比如24小时，需要用比特币付费。只要收到比特币支付，它就会一直保存客户文件。除了简单的存储，它还能做一些其他有趣的事情。我们不做深入展开。在我们的框架里，存储J实现了文件存取的去中心化，这也是像Dropbox[①]一样中心化服务的核心功能。代理其实就是中介。对我们来说，自动化与否其实并没有区别。然而，中介之间存在竞争。支付是通过比特币。代理提供服务和收到支付并没有不可分割的连接，所以安全

---

① 一款非常好用的免费网络文件同步工具，是Dropbox公司运行的在线存储服务，有人说是世界上最伟大的云存储服务之一。通过云计算实现互联网上的文件同步，用户可以存储并共享文件和文件夹。——译者注

是通过代理的声誉来实现的。

零币

我们在第6章讨论过零币，一个有效的、去中心化达到混合币种从而保持匿名服务的方法。零币不是采用中心化的混合服务，而是采用加密协议实现了用混合币种的功能，只是靠数学和共识而不靠中介。零币和它的后来者重度使用加密机制，因此比较适合使用特别的区块链。在安全措施上，我们曾提到需要销毁一个基础币（Basecoin）从而作为交换获取一个零币，这两个操作密不可分。赎回零币也是如此，这就是典型的不可分割性。

11.4 什么时候适合去中心化

我们一直在关注技术上如何实现去中心化。现在，我们要深入研究去中心化的动机。这些问题跟技术无关，却同样难以回答：去中心化一定是有益的吗？经济上合适吗？去中心化的社会影响是什么？

我们至今只是把“去中心化”当成一个技术概念，却从来没有明确提出去中心化是和政治紧密相关。当我们讨论用技术的创新手段替代或者部分替代传统体系，我们实际上是在讨论如何在旧有的法律、社会和金融机构中重新分配权力。去中心化的理想来自比特币的起源：密码朋克运动——一群不墨守成规的人发起的旨在通过加密技术实现个体自治的运动（参阅前言和第7章相关内容）。有了区块链，他们的理想相对以前更近了。但是他们的理想可行吗？受欢迎吗？

回到我们的汽车销售案例，传统机构会试图帮助汽车所有者解决两个问题：第一个是强化所有权，也就是说，防止偷窃；第二个是确保交易安全进行，或者说防止在交易中被欺骗。所以当我们比较智能资产和现有体系时，我们不能仅仅比较一切顺利的时候谁效率高，最重要的是我们要看事情不顺利的时候，最坏能坏到什么程度。

现实生活中的安全性挑战

防止任何形式的偷盗——比如汽车、艺术品、现金等——都是预防、发现和纠正错误。预防性安全机制在于阻止偷盗的发生，发现性安全机制在于发现盗窃便于将来采取纠正错误的措施，纠正错误就是索回盗窃的损失并惩罚盗窃者（惩罚也可以看成是预防盗窃发生的方式之一）。车锁和报警器是预防性安全措施，GPS 跟踪器（比如 LoJack）用于帮助发现盗窃并有助于执法者找到被偷的车。其中的关键在于，阻止偷车涉及警察、保险、法庭等，车锁只是阻止偷车犯罪的一个小因素。如果你生活在没有法律保护的环境里，单纯的车锁并不能阻止盗窃的发生。这种情况下，锁在路边的车很快就会被偷走。

智能资产的模式极大地依靠预防措施。只有把所有权等同于使用权，我们才能实现去中心化。即拥有这辆车所有权就表示，你知道某块区块链上的私人密钥。但是这种控制原理无法取代我们现在拼图式的机构合作。

假若我们仅用保密的私人密钥来表示拥有权，那么数字加密就变得很重要。但是因为人类是所有因素里最容易被突破的，数字加密的保密性就会变得很困难。加密系统的设计者几十年来一直在努力让非技术用户使用并管理私人密钥以防止偷盗或者丢失钥匙，但是毫无进展。如果去中心化只依靠私密钥匙，非法软件或者钓鱼攻击就能偷走车。或者忘记密钥就能导致你的车变成一堆废铁。当然，以上问题都可以有其他的解决方式，最后这些解决方式让我们回到中介和中心化系统，偏离了我们为之奋斗的去中心化的主旨。

只要涉及人，资产转移就会存在争议如何解决的问题。争议在于交易条件或者交易的任何一个方面。在现实世界，如果交易双方无法达成一致，就会把争议提交到法庭，法官会按照固定的程序验证每一条证据、供词和书面文件，最后法官会做出关于这个买卖有效性的判决。很多人，特别是技术方面的人，倾向于把法律看成是一套由逻辑化的规则与算法组成的体系，该体系必定能够得到一个明确的裁决。然而，现实中的法律体系不但有冗长的法律和条约，还有人们对法律法规的理解和判断。这些离有明确逻辑的法规的观念就更加遥远了。这种特征并不是弱点。因为这样才可能允许我们去解决当初制定法律的人

所没有想到的更加复杂的案例。

回到去中心化模式的安全问题，我们想要的安全属性和去中心化模式提供的安全模式是有差别的。就以之前提到的去中心化众筹为案例：我们知道众筹可以技术上设置成让创业者直到资助款达到一定数额才可以取出。然而，这并不能阻止已经成功筹集到钱的创业者携款潜逃。事实上，即使在现在中心化模式下，众筹网站也有一些骗局，最终被告上法庭。在去中心化模式下，创业者有可能会匿名，没有被法律诉讼的威胁，这种情况下骗局一定更多。很难想象技术能够解决这个难题。这是另一个案例，显示了技术只能解决一小部分问题。坦白地说，技术能解决的只是问题中无趣的那部分。

小结一下，智能资产的问题是社会性的，都是当事情不太对劲时引起的纠纷。在所有参与方都满意的前提下，技术可以保证交易的高效性。但是，技术无法解决这些棘手的纠纷。

智能资产的优缺点

正如我们讨论的，在一些传统上需要人们介入的方面，智能资产不容易去中心化。事实上，自动化可能让去中心化更加困难，因为调停和其他进程都是在事件发生后才出现，自动化很难协调好这种事后出现的不正常的事件。最后，智能资产产权会产生新的问题，比如在汽车的例子中，会要求软件和硬件都安全。

在某种程度上，我们讨论过的例子只是一个完整智能资产协议的缩印简化版。比特币世界中的很多提案都是更为周全与细致入微的。即便如此，在我们简单的案例中，我们还是可以分辨出智能资产的好处和坏处。

智能资产的最大优点，就是能够在任何时候、任何地点进行高效的所有权转让。对于像智能手机或者电脑这样的产品，价值没有汽车那么高，如果有争议一般也不会诉诸法律，所以使用智能资产没有任何坏处。对于这些产品，不可分割的交易是一种有用的安全性特性。

通过区块链的智能资产也可以提供很好的隐私保护甚至匿名服务。虽然我们说隐私保护会让解决争议变得复杂化，但对于一个消费者信息被公司滥用的

社会，隐私保护还是很有意义的。在某些情况下，不需要揭露真实身份是交易方最看重的因素。这在中心化的中介模式下是不可能的。

最后，去中心化的模式允许自由选择调停者。即使我们对法律系统很满意，大多数争议调停是由像维萨或者贝宝这样的私人公司内部暗箱操作的。通过使用可替代的模式，让调停公开竞争，我们有可能让这个过程更加透明公开并接受公众的监督。

加密、国家和大机会

崛起的现代国家和我们在本章讨论的技术的目的有惊人的相似。在从民族部落和小群体发展出来的现代社会，政府一直致力于精确地解决一个问题：让陌生人安心地开展商务和其他交互活动。政府和我们讨论的技术，采用的办法也许大不相同，但是目标是一致的。

尽管马克思主义者眼中的去中心化也许牵扯到解散国家，但解散国家这种做法不太现实，尤其是在其他人比如民主化国家的人民也希望去中心化时。然而，通过技术去中心化并不一定会和国家对立。事实上，它们可以相辅相成。比如，假设交易双方都是经过验证的，智能资产的交易就可以通过区块链技术高效地完成，万一有争议，也可以付诸法律。我们认为，未来区块链技术的机会就在于，以和政府功能互补的方式建立去中心化，而不是试图替代政府。

只要技术存在就可以去中心化，这种想法很吸引人。但是在实际操作中，我们需要找到经济上令人信服的理由，比如说政府的监管过于繁杂和低效，或者权力失衡导致的权力滥用。举个例子，很多非洲国家的居民已经用“手机分钟”（cell phone minutes）作为实际上的货币，这样的货币脱离了政府的控制，也避免了相应权力的滥用。

总结一下，我们在本章描绘了去中心化的技术蓝图，也批判性地探讨了去中心化的动因。我们鼓励大家寻找去中心化更有说服力的案例，特别是把现有法律和监管实践融合在一起的好案例。

CONCLUSION | 结束语

有些人因为其底层技术而对比特币感兴趣，另外一些人对它的商业可能性着迷，还有一些人关心它的社会和政治影响。理性的人可能不同意后面两类人的观点。但是我们希望这本书能够让你知道，比特币在技术上是有深度的、创新的、有趣的，而且是建立在正确理论上的。我们才刚刚开始开拓比特币之外令人炫目的另类加密货币，其中的某些加密货币也许有一天甚至超越比特币。

我们选择深入研究比特币，是因为我们坚信技术的力量。我们坚信比特币和其他计算机科学有很深的联系。我们重点突出了有潜力的新技术是如何试图取代已有组织机构的。我们相信，人类还会继续找到加密货币技术在新的商业和社会领域里的有益应用。即使你的兴趣主要在于它的商业化，你也能从了解掌握它的底层技术中获益。而了解它的能量和限制，有助于你在市场的浪潮起伏中顺应时势。

未来何往？去中心化的好处之一，就在于它是一个极佳的实验和学习的平台。任何人都可以下载安装和分析比特币的区块链，或者在此之上建立自己的应用。我们希望你也能充分利用这些机会。

我们制作了许多教材的网络辅助材料。Coursera 网上课程[①]（www.coursera.org/course/bitcointech）包含了根据本书录制的视频课程，还有测验和一些编程作业（在线资料链接为 http://press.princeton.edu/titles/10908.html）。参加这个网络课程还可以让你和志同道合的学习者一起在线讨论。

① 免费大型公开在线课程项目，由美国斯坦福大学两名计算机科学教授（Andrew Ng 和 Daphne Koller）创办。旨在同世界顶尖大学合作，在线提供免费的网络公开课程。Coursera 的首批合作院校包括斯坦福大学、密歇根大学、普林斯顿大学、宾夕法尼亚大学等美国名校。——译者注



术语表

Advanced Encryption Standard 高级加密标准（简称 AES）

altcoin infanticide 另类币夭折

Altcoin 另类币

anonymityset 匿名集

anonymous marketplace 匿名市场

Anti-Money Laundering 反洗钱（简称 AML）

append-only ledger 仅增账目

Application Programming Interface 应用程序编程接口（简称 API）

Application Specific Integrated Circuits 专用集成电路技术（简称 ASIC）

Arithmetic Logic Units 算术逻辑单元（简称 ALU）

ASIC-resistant puzzles 反 ASIC 解谜算法

asymmetric information 信息不对称

atomic cross-chain swaps 原子型交叉链互换

Basecoin 基础币

bent corner theory 折角论

Berkeley Open Infrastructure for Network Computing 伯克利开放式网络计算平台（简称 BOINC）

birthday paradox 生日悖论

bit fiddling 数位操作

bitcoin core 比特币核心钱包

Bitcoin Foundation 比特币基金会

Bitcoin Improvement Proposal 比特币改进方案（简称 BIP）

bitcoin mining 比特币挖矿

bitcoindlibrary 比特币官方客户端的资源库

bitcoin-qt library 比特币类库，现在又称为比特币中心（bitcoin core）

bitlicense 比特币牌照

block chain 区块链

blockchain. info 区块链信息公司

block-discarding attack 区块丢弃攻击

bootstrapping 自举过程

brain wallet 大脑钱包

bytecode 字节码

Byzantine Generals Problem 拜占庭将军问题

change address 零钱地址

chunk size 块大小

clustering of addresses 地址簇

CoiledCoin 盘旋币

coin center 货币中心

Coinbase 比特币基地公司

CoinJoin 合币

coinstake transaction 币拥有量交易

collision-resistance 碰撞阻力

collusion 串谋

Colored Coins 染色币

Commit Coin 承诺币

commitment 承诺

compatibility 兼容性

compression function 压缩函数

concensus chain 共识链

consensusalgorithm 共识算法

considating funds 资金合并

Counterparty 合约币

CreateCoins 造币

cryptocurrencyecosystem 加密货币生态系统

crypto-currency 加密数字货币

cryptographic beacons 密码学“信号塔”

cryptographic guarantees 加密学保证

Cunningham chain 坎宁安链

cypherpunk 密码朋克

Dark Coin 黑暗币

data furnace 数据火炉

deanonymized 暴露

decentralized mixing 分布式混币

default strategy 默认策略

digital cash 数字货币

digital signatures 数字签名

distributed consensus 分布式共识

distribution with high min-entropy 最小信息熵分布特性

Dogecoin 狗币

double spending 双重支付

ECDSA 椭圆曲线数字签名算法

efficient micro-payments 高效小额支付

encoding keys 编码解码

escrow transaction 第三方支付交易

Ethereum-specificVirtualMachine 以太坊专用虚拟机（简称 EVM）

Ethereum 以太坊

feather forking 羽量级分叉

Field-Programmable Gate Array 现场可编程门阵列（简称 FPGA）

Fischer-Lynch-Paterson impossibility result 不可能性结论

flooding algorithm 泛洪算法

flooping protocol 泛洪协议

forking attack 分叉攻击

frontrunning 预先交易

fully validating nodes 完全有效节点

fungibility 可互换性

Futurecoin 未来币

GenCoin 生成货币

genesis block 创世区块

getblocktemplate 获取有效区块模版（简称 GBT）

GoofyCoin 高飞币

Great Internet Mersenne Prime Search 互联网梅森质数大搜索（简称 GIMPS）

green addresses 绿色地址

Hash 哈希算法

hash collision 哈希碰撞

hash pointer 哈希指针

hash power 哈希算力

hash puzzles 哈希解谜

hash rate 哈希速度/哈希率

hashes of public keys 公钥哈希值

hiding 隐秘性

hierarchical deterministic wallet 分层确定性钱包

high min-entroy 高阶最小熵

high-level flows 高风险交易流

idioms of use 惯用法则

implicit consensus 隐性共识

instawallet 一种在线钱包

joint payments 共同支付

key stretching 密钥延展

key-value 键值

Know Your Customer 了解你的客户（简称 KYC）

Large Hadron Collider 大型强子碰撞（简称 LHC）

laundering hashes 洗算力

laundry 洗钱

lemons market 柠檬市场/次品市场

lightweight nodes 轻量节点

Linear Feedback Shift Registers 线性反馈移位寄存器（简称 LFSR）

linking 关联性

Litecoin 莱特币

lock_time 锁定时间

locking mechanisms 加锁机制

mandatory reporting 强制上报

megajoules 兆焦耳，百万焦耳

megawatt 兆瓦，百万瓦特

memory-bound puzzles 内存限制解谜

memory-hard mining puzzle 记忆储存体挖矿谜题

memory-hard puzzles 刚性内存解谜

memoryless process 无记忆进程的

merge avoidance 合并规避

mergemining 共同挖矿

Merkle trees 梅克尔树

Merkle-Damgard transform MD 变换

mining shares 挖矿工分

Mix net 混币网络

Mixing 混币

modular addition 模加法运算

multisignatures 多重签名

Namecoin 域名币

niche currency 利基货币

nonce 临时随机数

opcode 操作码

Open Computing Language 开放运算语言（简称 OpenCL）

open protocol 开放协议

OpenAseet 开放资产

open-source project 开源项目

open-source software 开源软件

open-source system 开源系统

orphan block 孤块

overlay currencies 附着币

parent node 父节点

partial hash-preimage puzzle 不完全哈希函数原像解谜

partial preimage 不完全原像

PayCoins 付币

paying for a proof 支付证明费用

pay-to-pubkey-hash 标准的比特币转账流程/支付到比特币地址的标准交易

Pay-to-script-hash 支付给脚本的哈希值（简称 P2SH）

Peercoin 点点币

Permacoin 永久币

Petabytes 拍字节（简称 PB）

Pigeonhole Principle 鸽巢原理

pool hopping 矿池跳换

Primecoin 质数币

private key 私钥

progress free 无关过程的

proof of burn 销毁证明

proof of Liabilities 负债证明

Proof of membership 隶属证明

Proof of non-membership 非隶属证明

proof of Reserve 准备金证明

proof of retrievabitlity 可恢复性证明

proof of storage 存储量证明

proof of “clairvoyance” 未来预测证明

proof-of-stake 权益证明

proof-of-workskiplist 工作证明跳表
proof-of-work 工作量证明
protein folding 蛋白质折叠
provision 准备金
pseudocode 伪代码
pseudonymity 化名
Pseudo-Random Generator 伪随机数发生器（简称 PRG）
publickey 公钥
pull requests 提交请求
Pump-and-dumpscams 拉高出货骗术
punitive forking 惩罚分叉
puzzle-friendliness 谜题友好
Quick Response code QR 码
race condition 竞态条件
radio telescope 射电望远镜
Random Acess Memory 随机存取存储器（简称 RAM）
randomness beacons 随机数“信号塔”
real scripts 实际脚本
reality keys 现实密钥
replace-by-fee 替代策略
reputation system 信誉评价系统
Request for Comments 评议请求（简称 RFC）
root 树根节点
Satoshi bones 中本聪骨头
Satoshi Dice 中本聪之骰
Satoshi Nakamoto 中本聪
save up 蓄力
scriptPubKey 输出脚本
scriptSig 输入脚本
ScroogeCoin 财奴币

Secure Hash Algorithm 256 安全哈希算法（简称 SHA－256）

secure timestamping 安全时间戳

selfish mining 自私挖矿

sidechains 侧链

sidechannels 旁路攻击

Simple Mail Transfer Protocol 简单邮件传输协议（简称 SMTP）

Simple Payment Verification 简单付款验证（简称 SPV）

smart contracts 智能合约

spare cycle 空闲周期

stack-based programming language 堆栈式编程语言

stake-grinding attacks 股权粉碎攻击

stratum 层

sybil attack 女巫攻击

taintanalysis 污点分析

tamper-resistant device 防损硬件

temporary block-withholding attacks 临时保留区块攻击

the 51 percent attack 51% 攻击

the head of list 链表头部

the nothing-at-stake problem 无利害关系问题

threshold cryptography 门限密码

threshold signature 门限签名

Tinkerbell effect 仙子效应

transaction syntax 交易语法

transaction graph analysis 交易图谱分析

tumbler 翻洗

uniform transactions 一致性交易

uniqueCoinID 唯一的货币编号

unlinkability 无关联性

vanity addresses 虚荣地址

virtual currency 虚拟货币

virtual mining 虚拟挖矿

zero confirmation transaction 零验证交易

Zerocash 零钞

Zerocoin 零币

zero-knowledge proof 零知识验证

Zetacoin 泽塔币



译者简介

帅初，毕业于中国科学院，从2013年起，就从事加密货币和区块链技术领域的开发和研究工作，具备丰富的区块链行业经验。现担任唯链科技（vechain）首席技术官，也是中国区块链开源平台QtumChain的设计者。

蔡凯龙，点石资产管理创始人，厦门抬钱论道资产管理公司执委会主席，互联网金融千人会联合创始人，百度支付海外顾问，恒生电子海外投资高级顾问。注册金融分析师（CFA），金融风险管理师（FRM），经济和计算机双硕士，金融博士生。曾任联想控股旗下P2P翼龙贷副总裁，互联网金融千人会执行秘书长，德意志银行（美国）战略科技部副总裁，美国能源公司MXEnergy风控经理，美国休斯顿大学商学院金融系助理教授。《金融时报》（中文版）、《新浪财经》等财经媒体的专栏作者，曾发表多篇关于互联网金融的文章。编辑出版《智慧众筹：互联网金融早餐会》一书。

许余洁，现任联合信用评级有限公司研究总监，中国资产证券化研究院首席研究员，西南财经大学特聘研究员。2013年7月起供职于中国证监会研究中心（2015年更名为中证金融研究院），暨证监会博士后科研工作站与清华大学五道口金融学院联合培养博士后。2014年从事明斯基《稳定不稳定的经济》一书中文版的翻译校稿工作。近年来，在《人民日报》、《金融法苑》、《中国金融》、《工业技术经济》、《中国经济报告》、《中国证券报》等报刊杂志上发表文章30余篇，并以笔名“余吉力”在财新博客上坚持撰写100篇读书心得，广为转载。

李耀光，中国人民大学财政金融学院经济学硕士，特许金融分析师（CFA），中国注册会计师（CPA）。现就职于某合资证券公司，担任结构融资总监，负责境内资产证券化、REITs 及结构化金融产品的设计与发行，并参与跨境证券化产品与 REITs 的研究或顾问工作，成功完成或执行的资产类型覆盖商业及工业地产、应收账款、银行信贷、消费金融、租赁资产、公共事业收费权等。在此之前，曾就职于某行业领先的内资证券公司资产管理部、四大国有银行总行，长期从事理财与资金池投资管理、结构化投融资相关工作，并担任中国资产证券化研究院研究员，中国资产证券化论坛理事及教育委员会委员。

高晓婧，毕业于北京外国语大学英汉同声传译专业；曾就职于华夏银行总行理财业务管理部门，现就职于兴业银行总行投资银行部；具有多年泛资管领域从业经验，目前主要研究领域包括：银行理财、泛资管及资产证券化业务。

洪浩，CFA，现任职于中泰证券债券与结构金融部，负责信贷资产证券化和企业资产证券化业务。曾任职于中国对外经济贸易信托有限公司，在信托公司建立了全流程的服务体系。负责或参与十余单公募、私募资产证券化项目。北京大学理学博士，中国资产证券化研究院特聘研究员。

邢早忠 / 金融时报社社长，中国资产证券化研究院理事长

金融科技会不会改变我们当前的社会和经济世界，甚至是全球政治格局？这是与当前变得火热的区块链技术相关的一个重要问题。对此，大家肯定会有多元化的视角和不同的答案。如果你也感兴趣，看看这本书中的研究会是很值得的。当然，阅读本书有一定的挑战，需要具备高深的密码学、计算机科学、经济学、金融学、博弈论以及网络科学等领域的基础知识。但是，金融从业人员中的精英，可以尝试这种智力挑战，通过区块链的去中心化模式设计，促进更多主体（节点）的"全民参与"，为谋求广大公众的利益重塑金融服务业。

霍学文 / 北京市金融工作局局长

区块链将成为金融科技的底层技术，它最有可能低成本地解决金融活动的信任难题。由于其加密算法特性，区块链将使金融发展进入算法金融时代。但是，它也需要规范，否则会成为做期货交易，甚至进行非法集资和高杠杆交易等违法或非法金融活动的来源。本书非常专业，系统地阐述了比特币和区块链技术的方方面面，并明确指出比特币与其他失败了的数字电子货币相比，区别在于比特币使用了区块链来实现去中心化。本书对于关心金融科技的市场和监管人员，以及理论研究人员均具有很好的参考价值。

刘信义 / 浦发银行行长

本书主要介绍了比特币在技术层面的运作模式，以及作为比特币实现去中心化的重要后台与技术工具——区块链。我们看到，一些基于区块链的科技创业公司已经开始慢慢地触及金融服务市场的痛点，它们为包括银行、保险在内的机构客户提供解决方案。区块链不会是银行终结的信号，区块链可以使它们寻找新的契机，从而得到新的发展机会。这本书为我们部分揭示了区块链等金融科技的具体应用、运行机制和可能的未来。

黄世忠 / 厦门国家会计学院院长

林华组织翻译的这本书，对比特币的运行机制和底层技术进行了最全面的阐述。这部普林斯顿大学教材原本是一组免费大型公开在线课程，该课程反响极好，学习者众多，原著作者为我们打造了一个极佳的实验和学习的平台，他们希望任何人都可以在互联网时代，根据自己的需要来学习，大家都可以下载安装和分析比特币的区块链，并在此基础上建立各自的应用，这完全符合区块链"去中心化"的理念。

唐　斌 / 深圳前海金融资产交易所总经理

基于区块链技术的电子货币正在颠覆我们的金融世界，而我们知道，比特币只是区块链技术的第一个重量级应用，现在已经有越来越多的应用构建在区块链技术之上。如果你对比特币的底层技术（密码学、计算机科学和区块链）感兴趣，或是对比特币在商业和社会领域的可能应用着迷，或者想了解比特币之外的令人炫目的另类加密货币，这本书都会是一个不错的选择。




ISBN 978-7-5086-6584-9

定价：79.00元